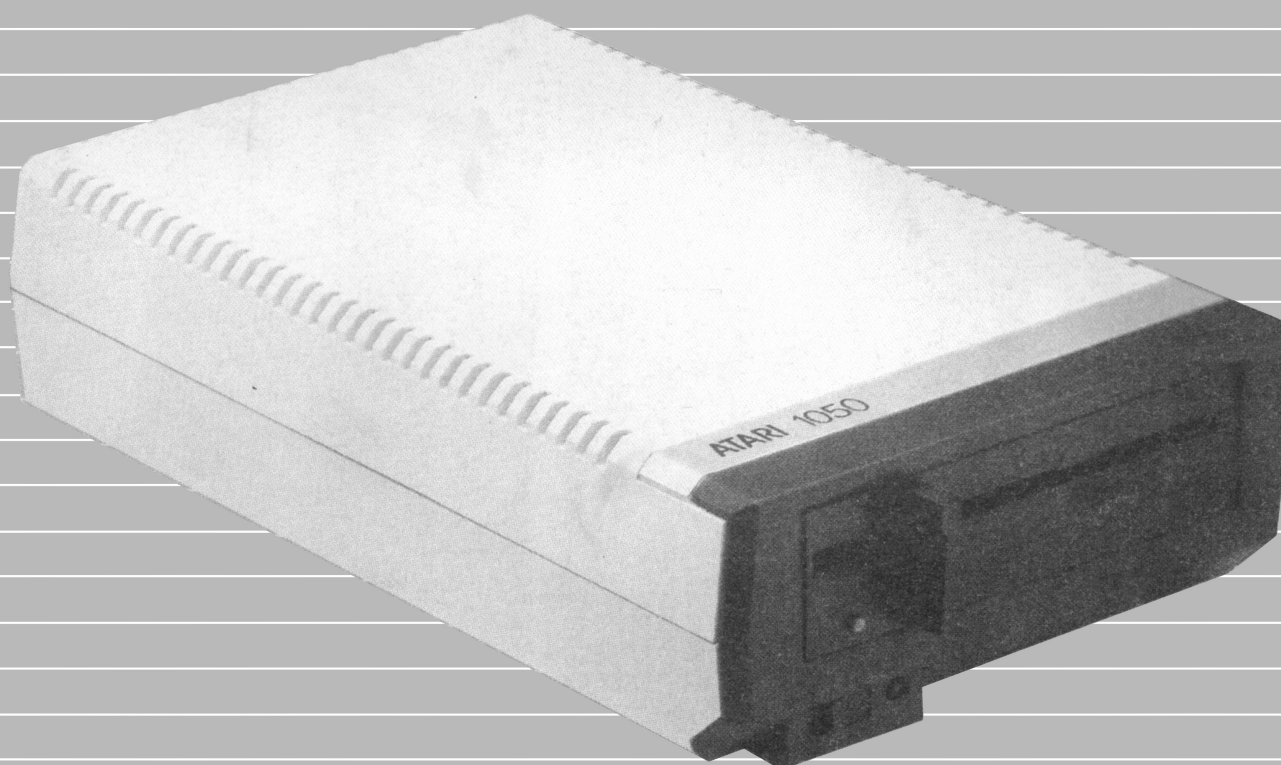
ATARI® 1050™
UNITÀ A DISCHI

MANUALE PER
SISTEMA OPERATIVO DOS II

NOTA AI POSSESSORI DELL'UNITÀ A DISCHI ATARI 1050

I riferimenti all'unità a dischi ATARI 810 in questo manuale sono direttamente applicabili all'unità a dischi ATARI 1050. Quando questo manuale si richiama alla "ATARI 810: guida per l'utente" consultate quest'ultima che viene fornita con la vostra unità dischi ATARI 1050.

Il DOS II e l'unità a dischi ATARI 1050 lavorano sia con il nuovo home computer ATARI 800XL che con gli home computers ATARI 400 e ATARI 800. Ignorate tutti i riferimenti all'unità a dischi doppia ATARI 815.

MANUALE DEL SISTEMA OPERATIVO A DISCHI II

PREFAZIONE

Il manuale del sistema operativo a dischi II (DOS II) ATARI è stato strutturato per due livelli di utenti. I nuovi utenti troveranno il DOS II spiegato con termini precisi e con le istruzioni su come procedere. Tuttavia i nuovi utenti del sistema dovrebbero prima:

- leggere il manuale dell'ATARI 400 o dell'ATARI 800 e 800XL;
- leggere il manuale dell'unità a dischi ATARI 810 o dell'unità a dischi doppia ATARI 815;
- acquisire una certa familiarità con uno dei linguaggi di programmazione ATARI (preferibilmente BASIC).

Tutti i nuovi utenti acquisteranno familiarità con il DOS II nei capitoli da 1 a 4, potranno quindi esplorare i capitoli 5 e 6 che sono progettati per gli utenti più esperti.

L'utente più esperto è colui che ha lavorato con l'ATARI DOS I (versione 24/9/79) o ha una conoscenza approfondita dell'ATARI Personal Computer System, del BASIC ATARI, e possibilmente dell'assembly e di altri linguaggi di programmazione.

Per entrambi i livelli, le spiegazioni sono accompagnate, per quanto possibile, da illustrazioni. La nostra intenzione è di presentare il materiale nella forma più chiara e concisa possibile per gli utenti di entrambi i livelli.

Le note e la terminologia usata con il DOS II possono essere trovate nella Appendice B; un glossario dei termini è disponibile nella Appendice J; l'uso delle istruzioni BASIC con il DOS II è spiegato nel dettaglio del capitolo 5.

Gli utenti esperti dovrebbero leggere l'Appendice I che spiega le principali differenze fra DOS I e DOS II, notando specialmente l'aggiunta della funzione MEM. SAV.

INDICE

COME USARE QUESTO MANUALE

Questo manuale è stato sviluppato tenendo sempre presente l'utente. Ogni sezione di informazione rappresenta una fase della seconda versione del sistema operativo a dischi ATARI (DOS II). Chi si accosta per la prima volta al DOS può trovare facilmente le informazioni necessarie ad iniziare ad operare col DOS II senza sentirsi oppresso da informazioni inutili. L'utente esperto può, tuttavia, trovare ed usare velocemente le istruzioni necessarie per compiere operazioni più complesse.

PER I NUOVI UTENTI DEL DOS

Il capitolo 1 spiega come usare questo manuale e le procedure per le operazioni più semplici:

- Definizione di DOS
- Installazione del sistema
- Spiegazione del Menu DOS

I capitoli 2 e 3 trattano e spiegano tutto ciò che riguarda i dischetti, dal formattamento alla memorizzazione. I principianti del DOS dovrebbero leggere i capitoli 1 e 2 prima di mettersi al lavoro. Ciò vi darà l'opportunità di acquistare familiarità con quanto vi accingete a fare.

PER I PRINCIPIANTI E PER I PIÙ ESPERTI

Il capitolo 3 spiega come identificare i vostri file su dischetto usando il filename (nomi dei file) e i filename extender (estensioni dei nomi dei file); questo capitolo vi introduce, inoltre, al caricamento e al salvataggio dei programmi.

Il capitolo 4 è il punto di incontro fra gli utenti nuovi e quelli più esperti e dà una descrizione dettagliata di ogni opzione del Menu DOS e di come usarla. Alcune di queste opzioni interesseranno solo gli utenti che hanno familiarità con il cartridge Editor Assembler e il sistema esadecimale.

PER GLI UTENTI PIÙ ESPERTI

Il capitolo 5 riprende le istruzioni BASIC usate con il DOS II e presenta dei programmi tipo che mostrano l'uso delle istruzioni I/O. Ogni tipo di istruzione è inoltre seguita da un esempio che mostra i tipi di dati da inserire in ogni parametro. Il capitolo 6 contiene ulteriori informazioni riguardo l'unità a dischi ATARI e ai dischetti (che possono interessare entrambi i livelli di utenti), assieme a dettagli che riguardano la memorizzazione e il reperimento dei dati.

Il manuale contiene, infine, un glossario dei termini e altre informazioni per gli utenti avanzati, del tipo:

- Mappa di memoria
- Errori
- Come risparmiare spazio RAM.

1

COMINCIAMO AD USARE IL DOS II

DOS (pronunciato doss) è un acronimo per: Disk Operating System (Sistema operativo a dischi). Senza un DOS, il vostro personal computer ATARI non può funzionare con l'unità a dischi ATARI 810. Il DOS consiste in una serie completa di routine che vi permettono di:

- Memorizzare programmi sui dischetti
- Richiamare programmi dai dischetti
- Creare e aggiungere dati ai file usati dai programmi
- Fare copie dei file
- Cancellare vecchi file dai dischetti
- Caricare e salvare file binari (per gli utenti avanzati)
- Muovere file nella e dalla memoria, lo schermo, i dischetti e la stampante.

Se non avete mai usato un DOS, troverete l'ATARI DOS II semplice da capire e da usare. Se avete già usato l'ATARI DOS I, troverete differenze nel caricamento del DOS II e cambiamenti sia nelle opzioni del Menu che nei parametri delle loro istruzioni. Questi cambiamenti rendono il DOS II più efficiente poiché avete più memoria disponibile e una maggiore flessibilità che non con la versione 24/9/79 del DOS I (vedere appendice H).

INSTALLAZIONE

Per iniziare dovete collegare la vostra unità ATARI 810 o la doppia unità a dischi ATARI 815 al vostro Personal Computer ATARI 400 o ATARI 800 assicurandovi che il computer abbia almeno 16 K di memoria ad accesso casuale (RAM). Per dettagliate istruzioni sull'installazione delle vostre apparecchiature, fate riferimento all'"ATARI 400 (o ATARI 800) Operator's Manual" (Manuale per l'operatore dell'ATARI 400 o dell'ATARI 800). Il procedimento per collegare l'unità a dischi doppia ATARI 815 o l'unità a dischi ATARI 810 al vostro personal computer è spiegato nei rispettivi manuali per l'operatore dell'unità a dischi.

AGGIUNTA DI PIÙ UNITÀ A DISCHI

È possibile collegare unità a dischi addizionali al vostro personal computer. Sul retro di ogni unità a disco ATARI vi sono due prese (chiamate I/O PERIPHERAL); collegate il cavo I/O dalla vostra seconda unità a dischi al drive (unità a dischi) 1. Quindi collegate l'altro cavo dal drive 1 alla console del computer. La figura 1-1 mostra alcune delle diverse configurazioni disponibili.

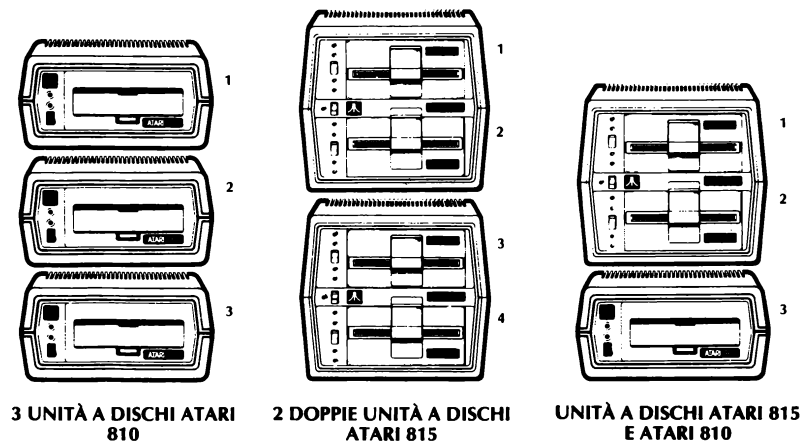



Figura 1-1 Configurazioni disponibili delle unità a dischi

POSIZIONAMENTO DEI CODICI SULLE UNITÀ

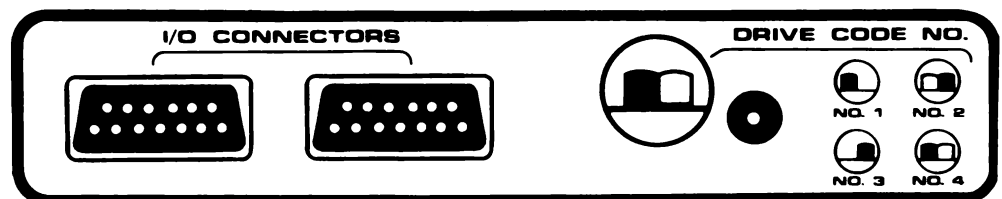
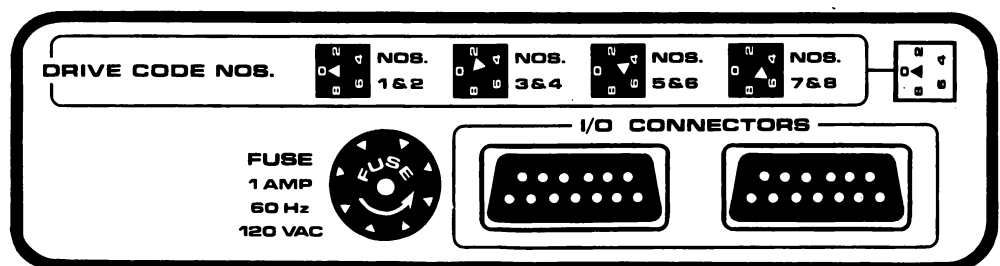
Guardando il retro della vostra unità a dischi ATARI 810, noterete un foro con due linguette. Le linguette bianca e nera devono essere messe in relazione al numero assegnato a quella unità. Riferitevi alla figura 1-2 a) per i codici opportuni. Il retro della vostra unità doppia a dischi ATARI 815 ha un disco che dovete ruotare nella posizione corretta per stabilire i codici opportuni (vedere la figura 1-2 b). Se avete solo un'unità doppia a dischi ATARI 815 il drive superiore è già messo a 1 e l'inferiore a 2. È importante assicurarsi che ogni drive abbia un codice diverso.

(a) Unità a dischi ATARI 810

(b) Unità a dischi doppia ATARI 815

Figura 1-2 Posizionamento dei codici delle unità

COME ETICHETTARE LE VOSTRE UNITÀ A DISCHI

Dopo aver stabilito i codici dei drives, etichettate ogni unità a dischi con il proprio numero onde evitare errori durante il loro uso. È essenziale che il vostro dischetto principale (Master diskette) o il disco di sistema sia SEMPRE posto nell'unità a dischi 1.

INSERIMENTO DI UN DISCHETTO

L'inserimento di un dischetto in una unità a dischi ATARI è una operazione semplice ma estremamente importante. Se il dischetto è posizionato in modo improprio, si possono avere errori d'"inizio" (Boot Errors) durante il procedimento di caricamento del DOS e il dischetto può rimanere danneggiato. Accendete la (le) unità a dischi e attendete finché la spia BUSY si spegne. Inserite il dischetto come segue:

1. Togliete il dischetto dalla sua busta di carta.

   Avvertenza: prendete il dischetto SOLO per il suo contenitore nero sigillato. NON toccate la parte di superficie esposta del dischetto, poiché ciò potrebbe danneggiare o annullare la sua capacità di lettura/scrittura. NON tenete il dischetto mettendo le dita sul foro centrale. NON cercate di rimuovere il dischetto dal suo involucro sigillato nero.

2. Tenete il dischetto con l'etichetta rivolta in alto e verso di voi e con la freccia sull'etichetta puntata verso lo sportello dell'unità a dischi (vedere figura 1-3). Inoltre, se il dischetto ha una tacca di protezione, questa dovrebbe rimanere alla vostra sinistra. Si noti che il vostro dischetto principale non ha la tacca di protezione perché viene automaticamente protetto alla fabbrica.

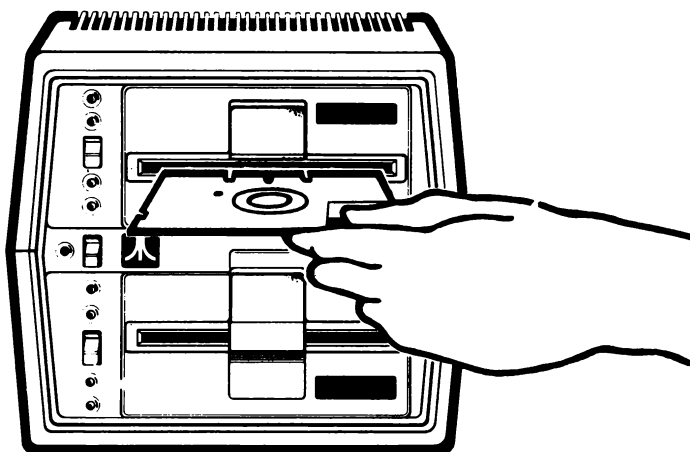



Figura 1-3 Inserimento di un dischetto nell'unità a dischi

3. Aprite lo sportello dell'unità a dischi (di quella superiore, se avete un'unità a dischi doppia ATARI 815) e delicatamente ma con fermezza fatevi entrare il dischetto.

4. Chiudete lo sportello dell'unità a dischi.

CHE COS'È UN MENU DOS?

Il Menu DOS è caricato nel vostro computer dal dischetto principale che è sempre inserito nell'unità a dischi N. 1. Il Menu DOS è simile al menu di un ristorante; una selezione di funzioni vi è presentata attraverso lo schermo del televisore. Fate la vostra scelta battendo la lettera di codice corrispondente e premendo **RETURN**. La funzione che avete selezionato è ora pronta all'uso.

Le selezioni del Menu DOS II sia per l'unità a dischi ATARI 810 (densità singola, 128 bytes per settore) che per l'unità doppia a dischi ATARI 815 (densità doppia, 256 bytes per settore) sono identiche. L'unica differenza è il nome della versione sullo schermo, come spiegato più sotto:

- Densità singola (usata sull'unità a disco ATARI 810): vedrete "Versione 2.OS" nella parte in alto a destra dello schermo. S sta per densità singola.
- Densità doppia (usata sull'unità doppia a dischi ATARI 815): vedrete "Versione 2.OD" nella parte in alto a destra dello schermo. D sta per densità doppia.

COME CHIAMARE IL MENU DOS?

Dopo aver acceso il televisore, inserito il dischetto principale nell'unità a dischi e acceso la vostra unità a dischi, siete pronti a caricare il DOS II nella memoria del computer come segue:

CON UN CARTRIGE INSTALLATO

Quando accendete il vostro computer:

1. La spia BUSY resterà accesa durante il processo di caricamento. NON cercate di rimuovere il dischetto quando questa spia è accesa. Se avete inserito il cartridge ATARI BASIC, un "READY" apparirà sullo schermo non appena il DOS II è stato caricato. (Se avete inserito il cartridge Editor Assembler, apparirà EDIT). Questo completa la prima parte della procedura di caricamento.
2. Battete **DOS** e premete **RETURN**. Il menu DOS II verrà mostrato sullo schermo (vedere figura 1-4).
   Questo completa la seconda parte della procedura di caricamento.

```
DISK OPERATING SYSTEM II VERSION 2.0S
COPYRIGHT 1980 ATARI

A. DISK DIRECTORY      I. FORMAT DISK
B. RUN CARTRIDGE       J. DUPLICATE DISK
C. COPY FILE           K. BINARY SAVE
D. DELETE FILE(S)      L. BINARY LOAD
E. RENAME FILE         M. RUN AT ADDRESS
F. LOCK FILE           N. CREATE MEM.SAV
G. UNLOCK FILE         O. DUPLICATE FILE
H. WRITE DOS FILES

SELECT ITEM OR RETURN FOR MENU
```

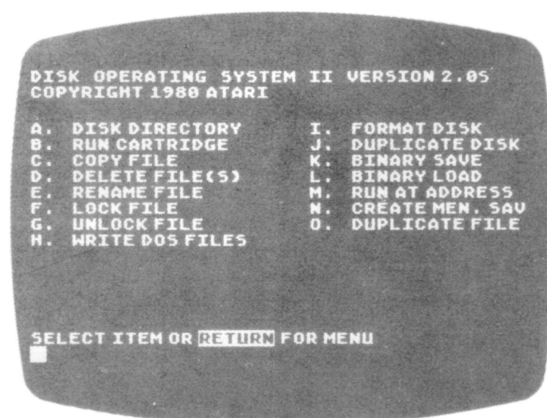



Figura 1-4 Il Menu DOS II

SENZA CARTRIDGE INSTALLATI

Quando accendete il vostro computer, il DOS II sarà caricato interamente senza interventi dell'utente. La spia BUSY resterà accesa durante il processo di caricamento. NON cercate di rimuovere il dischetto mentre questa spia è accesa. Il Menu DOS verrà mostrato automaticamente non appena il DOS II è caricato.

Dopo aver scelto l'opzione desiderata, un messaggio apparirà sullo schermo richiedendovi informazioni. Questo, il più delle volte sarà:

SELECT ITEM OR RETURN FOR MENU

(Selezionate la voce o battete **RETURN** per tornare al Menu)

Questo significa che il calcolatore attende che facciate una di queste operazioni:

- Battere una delle lettere e quindi **RETURN** per fare la selezione
- Premere **RETURN**, il che mostra di nuovo il Menu DOS.

LE OPZIONI DEL MENU DOS II

Di seguito sono mostrate le opzioni del Menu DOS II e una breve spiegazione di ciò che fanno. Vi consigliamo di NON usare queste opzioni finché non le capirete completamente (una spiegazione dettagliata appare nella sezione 4)

DISK DIRECTORY

Contiene una lista dei file contenuti su un dischetto. Se selezionate l'opzione **A** e premete **RETURN** DUE VOLTE, appariranno i nomi dei file e le estensioni (se ve ne sono), il numero dei settori occupati dal file e il numero di settori ancora liberi.

RUN CARTRIDGE

(Può essere usato SOLO con un cartridge installato nella console del computer). Questo vi permette di passare il controllo del sistema al cartridge inserito nella apertura che si trova a sinistra nel personal computer ATARI 800.

COPY FILE

Usate questa opzione quando avete due o più unità a dischi e volete copiare i file da un dischetto a un'altro. Usate questa opzione anche quando volete due file con le stesse informazioni sullo stesso dischetto, assegnando un secondo nome al file originale.

DELETE FILE

Questa opzione vi permette di cancellare un file dal dischetto, aumentando lo spazio disponibile.

RENAME FILE

Usate questa opzione quando volete cambiare il nome di un file.

LOCK FILE

Questa opzione vi premunisce dal modificare o cancellare accidentalmente il file. Sarete ancora in grado di leggere il file ma non di scriverci sopra. Per indicare un file "bloccato" (locked) viene posto un asterisco davanti al nome del file quando questo appare sullo schermo.

UNLOCK FILE

Rimuove l'asterisco dal nome del file e vi permette di fare modifiche e cancellazioni sul file.

WRITE NEW DOS

Usate questa opzione per sostituire o aggiungere i file DOS (DOS.SYS e DUP.SYS) che si trovano sul vostro dischetto principale, su un dischetto in qualsiasi unità a disco.

FORMAT DISKETTE

Questa opzione è usata per formattare un dischetto nuovo, il che è necessario prima di poter scrivere qualsiasi cosa su di esso. Assicuratevi di non avere alcun file sul dischetto prima di formattarlo. Questa opzione non è normalmente usata su un dischetto formattato ATARI 810 (CX8111).

DUPLICATE DISK

Questa è l'opzione da scegliere quando volete creare un esatto duplicato di un dischetto (per informazioni più dettagliate fate riferimento al capitolo 2 sulla duplicazione dei dischetti).

DUPLICATE FILE

Questa opzione vi permette di copiare un file da un dischetto ad un altro anche se avete una sola unità a dischi (vedere il capitolo sulla duplicazione dei file per maggiori dettagli).

CREATE MEM.SAV

Questa opzione vi permette di creare settori di spazio disponibili sul dischetto per il programma in RAM che deve essere salvato mentre è stato usato il file DUP.SYS. (Per ulteriori informazioni vedere: CREATE MEM.SAV nella sezione 4). Vi raccomandiamo di creare un file MEM.SAV in ogni nuovo dischetto che intendete usare come dischetto di sistema. Impratichendovi maggiormente con il DOS, noterete che vi sono casi in cui un file MEM.SAV non serve. Pertanto l'inconveniente di dover aspettare parecchio tempo prima che il file sia caricato in memoria può autorizzarvi a cancellarlo dal dischetto. Un caso del genere si presenta quando non ci sono programmi in RAM che volete proteggere quando caricate il DOS.

BINARY SAVE*

Con questa opzione potete salvare il contenuto di locazioni di memoria specifiche su un dischetto (tratta i programmi in linguaggio Assembly).

BINARY LOAD*

Questa opzione vi permette di caricare in memoria un file da un dischetto. È la funzione inversa di BINARY SAVE (tratta i programmi in linguaggio Assembly).

RUN AT ADDRESS*

Con questa opzione potete introdurre l'indirizzo esadecimale di inizio di un programma in oggetto dopo che è stato caricato in RAM con una BINARY LOAD (esegue programmi in linguaggio Assembly).

Nota: BINARY SAVE, BINARY LOAD e START AT ADDRESS sono per gli utenti avanzati del DOS II e sono spiegate molto dettagliatamente nel capitolo 4°

2

I DISCHETTI

IL DISCHETTO PRINCIPALE (MASTER DISKETTE)

Il dischetto principale contiene i programmi del sistema operativo. Questi programmi includono tutte le gestioni dei file del sistema e routine d'utility, necessarie per far funzionare l'unità a dischi con il vostro personal computer ATARI. Senza un sistema operativo a dischi non potete accedere all'unità a dischi.

Ogni dischetto principale contiene i seguenti file:

DOS.SYS File

DOS.SYS è un file contenente il sottosistema per la gestione dei file (File Manegement Subsystem – FMS) e la porzione residente in RAM del DUP.SYS (spesso chiamata "mini DOS"). La porzione residente in RAM del DUP.SYS contiene le sottofunzioni che possono essere controllate dal FMS: DELETE FILE, RENAME FILE, LOCK FILE, UNLOCK FILE e FORMAT DISK.

DUP.SYS File

È l'insieme d'utility su disco (Disk Utility Package) che contiene il Menu DOS e le sottofunzioni del DOS NON controllate dal FMS. Ogniqualvolta volete vedere il Menu DOS o scegliere queste sottofunzioni DOS: BINARY LOAD, BINARY SAVE, RUN AT ADDRESS, RUN CARTRIDGE, COPY FILE, DUPLICATE FILE e DUPLICATE DISK, dovete caricare il DUP.SYS file nella RAM battendo **DOS** e premendo **RETURN**.

Nota: normalmente quando portate il DUP.SYS file nella RAM esso cancella i dati nella parte bassa dell'area del programma occupata da programmi nei linguaggi BASIC o Assembly. Tuttavia, quando create un file MEM.SAV (vedere il capitolo sul MEM.SAV) sul vostro dischetto, il mini DOS salva i dati in RAM sul dischetto prima di caricare DUP.SYS. Quando avete terminato di usare le funzioni DUP-.SYS, MEM.SAV vi permette di ricaricare automaticamente il programma.

AUTORUN.SYS File

Questo file è usato per controllare le unità periferiche (se ve ne sono) collegate al vostro personal computer ATARI e caricare i programmi in codice macchina (vedere il capitolo 4 di questo manuale per ulteriori dettagli sul AUTORUN.SYS).

I DISCHETTI FORMATTATI ATARI 810

In aggiunta al dischetto principale II (CX8104) la vostra unità a dischi ATARI 810 viene fornita con un dischetto formattato II ATARI 810 (CX8111). Questo dischetto non contiene file o programmi ed è stato formattato in produzione; ciò significa che il dischetto è già stato diviso in piste e settori (vedere fig. 2-1) il che rende possibile la memorizzazione e il richiamo di informazioni più rapidamente di come avverrebbe con dischetti formattati sulla vostra unità a dischi ATARI 810. Questo dischetto "vuoto" è fornito affinché possiate fare una copia del dischetto principale.

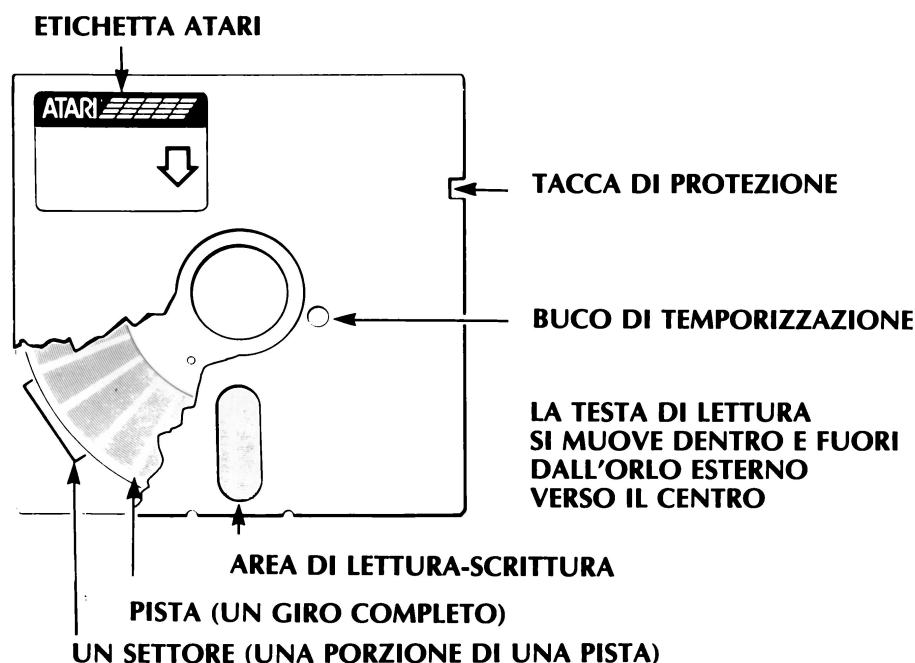



Figura 2-1 Un dischetto formattato

Questa copia, chiamata dischetto di sistema, è quella con la quale lavorerete effettivamente per evitare di danneggiare il dischetto principale originale (vedi come ottenere un dischetto di sistema).
Se, comunque, decidete di cancellare i file da un dischetto preformattato, usate l'opzione D del Menu DOS: DELETE FILE(S). Questa operazione non comprometterà la maggiore velocità del dischetto preformattato. Se, per qualsiasi ragione, riformattate il dischetto, questo perderà la maggiore velocità caratteristica di un dischetto preformattato.

Nota: con l'unità a dischi ATARI 810 potete anche usare dischetti nuovi non preformattati in produzione. Dovrete formattare questi dischetti usando l'opzione del Menu FORMAT DISK prima di potervi scrivere il DOS file o memorizzare programmi (vedi come formattare un dischetto). Tuttavia, non avrete il vantaggio della maggiore velocità dei dischetti preformattati.

COME FORMATTARE UN DISCHETTO

Dovrete formattare qualsiasi dischetto nuovo prima di poter scrivere su di esso (a meno che non stiate usando un dischetto preformattato II ATARI 810 (CX8111)). Formattandolo, il dischetto viene organizzato in modo che il DOS II sappia dove sono localizzate le informazioni. A differenza di un disco di musica che ha incisi dei solchi visibili, il dischetto ha dei "solchi" magnetici invisibili.

Dopo aver formattato un dischetto, esso conterrà 40 di queste piste magnetiche che sono divise in 18 settori. La capacità di memorizzazione si ottiene moltiplican-

do il numero di piste (40) per il numero di settori in ogni traccia (18), cioè 720 settori. Tuttavia, 13 dei 720 settori sono utilizzati dal DOS II e non sono disponibili per scrivervi file e dati. L'effettiva suddivisione è mostrata di seguito:

3	settori usati all'accensione del sistema
8	settori usati dalla lista dei file (directory)
1	settore usato per indice
1	il settore 720 non è indirizzabile
13	totale

Come risulta, sono effettivamente disponibili 707 settori nei quali è possibile scrivere i dati. L'unità a dischi a densità singola ATARI 810 può memorizzare 128 bytes di informazioni in ogni settore del dischetto. Siccome 3 bytes per settore sono riservati al Sottosistema per la gestione dei file (FMS), la capacità di memoria totale è di 88.375 bytes.

Per formattare un dischetto usate l'opzione I (FORMAT DISK) del Menu DOS. Ciò non è necessario se state usando un dischetto formattato II ATARI 810 (CX8111). Se state usando una unità a dischi ATARI 810 come unità 1 consultate le istruzioni (A) più sotto. Se state usando una unità a dischi doppia ATARI 815 come unità 1 e 2 consultate le istruzioni (B).

(A) Usando l'unità a dischi ATARI 810

1. Accendete l'unità a dischi e attendete che la spia BUSY sia spenta.

2. Assicuratevi che gli interruttori sul retro dell'unità siano posti a 1 (fate riferimento alla figura 1-2 per la messa a punto dei codici sui drive).

3. Inserite il dischetto principale (Master Diskette II) ATARI 810 (CX8104) e chiudete lo sportello del drive.

4. Accendete la console del computer. Il DOS sarà caricato nella memoria del computer.

5. Quando appare il messaggio READY (se avete il cartridge ATARI BASIC inserito) battete **DOS** e premete **RETURN**. Dopo pochi secondi apparirà sullo schermo il Menu DOS (se non sono inseriti cartridge, il Menu DOS apparirà automaticamente).

6. Battete **I** per l'opzione FORMAT e premete **RETURN**.

7. Quando appare il messaggio WHICH DRIVE TO FORMAT (quale drive per formattare) rimuovete il dischetto principale dalla unità a dischi e inserite un dischetto BLANK (vuoto). Chiudete lo sportello, battete **1** e premete **RETURN**.

8. Quando appare il messaggio TYPE Y TO FORMAT DISK 1 (batti Y per formattare il disco 1) batti **Y** e premi **RETURN**. La spia BUSY si accenderà e il sistema formatterà il dischetto.

9. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU (seleziona una opzione o Return per il Menu) il formattamento è completo e potete scrivere file su quel dischetto.

Se avete due o più unità a dischi ATARI 810 potete formattare un dischetto blank su ognuna di queste; dovete conoscere il codice di quel drive così da poter rispondere al messaggio WHICH DRIVE TO FORMAT.

(B) Usando una unità doppia a dischi ATARI 815

1. Accendete le unità a dischi e attendete che la spia BUSY si spenga.

2. Assicuratevi che gli interruttori sul retro delle unità a disco siano posizionati correttamente (fate riferimento alla figura 1-2 per la messa a punto dei codici dei drive).

3. Inserite il dischetto principale (Master Diskette II) ATARI 815 (CX8201) nel drive 1 e chiudete lo sportello del drive.

4. Accendete la console del computer. Il DOS sarà caricato nella memoria del computer.

5. Quando appare il messaggio READY (se avete il cartridge ATARI BASIC inserito) battete **DOS** e premete **RETURN**. Dopo alcuni secondi il menu DOS apparirà sullo schermo. (Se il cartridge non è inserito, il Menu DOS apparirà sullo schermo automaticamente).

6. Battete **I** per l'opzione FORMAT e premete **RETURN**.

7. Quando appare il messaggio WHICH DRIVE TO FORMAT (quale drive per formattare) mettete il dischetto blank nel drive 2, chiudete lo sportello e battete **2**, quindi premete **RETURN**.

8. Quando appare il messaggio TYPE Y TO FORMAT DISK 2 (batti Y per formattare il disco) battete **Y** e premete **RETURN**. Si accenderà la spia BUSY e il sistema formatterà il dischetto nel drive 2.

9. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU (selezionate un'opzione o Return per il Menu) il formattamento è completo e potete scrivere i file su quel dischetto.

COME OTTENERE UN DISCHETTO DI SISTEMA DAL VOSTRO DISCHETTO PRINCIPALE

La prima operazione che dovete fare è duplicare il vostro dischetto principale. Si fa questo per proteggere il dischetto principale da ogni danno accidentale; il duplicato del vostro dischetto principale sarà chiamato Dischetto di Sistema (copia di lavoro) ed è quello che userete normalmente per caricare il DOS nella RAM.

Il modo di procedere per questa operazione dipende dal tipo di unità a dischi che usate.

CREARE UN DISCHETTO DI SISTEMA CON L'UNITÀ A DISCHI ATARI 810

1. Accendete il televisore, accendete l'unità a dischi e attendete che la spia BUSY (la luce rossa in alto) si spenga.

2. Togliete il dischetto principale dalla sua custodia di carta.

3. Inseritelo nell'unità a dischi e chiudete lo sportello.

4. Accendete la console del computer.

5. Supponendo che abbiate un cartridge BASIC inserito nella console, vedrete un messaggio READY. Battete **DOS** e premete **RETURN** (se non avete il cartridge inserito nella console, il Menu verrà caricato automaticamente).

6. Togliete il dischetto principale e inserite un dischetto formattato nell'unità a dischi. Questo dischetto vuoto può essere:

   - un dischetto formattato II ATARI 810 (CX8111)
   - un dischetto che avete formattato in precedenza usando il DOS II
   - un dischetto che avete riformattato usando il DOS II.

7. Battete **H** e premete **RETURN** per l'opzione WRITE DOS FILES (scrivi i file DOS).

8. Quando appare il messaggio DRIVE TO WRITE DOS FILES TO? (su quale drive scrivo i file DOS?) battete **1** e premete **RETURN**.

9. Quando appare il messaggio TYPE Y TO WRITE DOS TO DRIVE 1 (batti Y per scrivere il DOS sul drive 1) battete **Y** e premete **RETURN**.

10. Apparirà sullo schermo il messaggio: WRITING NEW DOS FILES (scrittura dei nuovi file DOS in esecuzione).

11. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU il dischetto principale è stato duplicato e avete creato un dischetto di sistema.

    Nota: A questo punto vi consigliamo caldamente di creare un file MEM.SAV (vedi il paragrafo sul MEM.SAV in questo capitolo). MEM.SAV riserva un certo numero di settori sul dischetto di sistema (e sugli altri dischetti) per memorizzare il programma RAM residente mentre usate le funzioni DOS, cioè, il file DUP.SYS (vedere il capitolo che spiega la funzione DUP.SYS).

12. Battete **N** e premete **RETURN** per creare un file MEM.SAV sul vostro dischetto di sistema.

13. Quando appare il messaggio TYPE Y TO CREATE MEM.SAV (battete Y per creare un file MEM.SAV) battete **Y** e premete **RETURN**.

14. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU, sul vostro dischetto di sistema vi è ora un file MEM.SAV.

Se avete due o più unità a dischi ATARI 810 potete inserire il vostro dischetto formattato in ognuno di questi prima di scegliere l'opzione H del Menu (WRITE DOS FILES). Dovete però ricordare quale drive state usando così da poter rispondere alla domanda DRIVE TO WRITE DOS FILES TO? al passo 8.

CREARE UN DISCHETTO DI SISTEMA USANDO L'UNITÀ DOPPIA A DISCHI ATARI 815

1. Accendete l'unità doppia a dischi ATARI 815. Assicuratevi che il codice del drive sul retro sia stato posizionato in modo corretto (vedere figura 1-2). Il drive superiore sarà il numero 1 e l'inferiore il numero 2.

2. Togliete il dischetto principale dalla sua custodia di carta.

3. Inseritelo nell'unità a dischi e chiudete lo sportello.

4. Accendete la console del computer.

5. Se avete un cartridge BASIC inserito nella console, vedrete un messaggio READY. Battete **DOS** e premete **RETURN** (se non avete il cartridge inserito nella console, il Menu verrà caricato automaticamente).

6. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU battete **H** e premete **RETURN** per selezionare l'opzione WRITE DOS FILES (scrivi i file DOS).

7. Quando appare il messaggio DRIVE TO WRITE FILES TO (quale drive usare per scrivere i file) mettete un dischetto formattato nel drive 2 e chiudete lo sportello, quindi battete **2** e poi **RETURN**.

8. Quando appare il messaggio TYPE Y TO WRITE DOS TO DRIVE 2 (premi Y per scrivere il DOS col drive 2) premete **Y** e quindi **RETURN**.

9. Apparirà il messaggio WRITING NEW DOS FILES (scrittura dei nuovi file DOS in esecuzione).

10. Quando appariranno il Menu DOS e il messaggio SELECT ITEM OR RETURN FOR MENU, avrete creato un dischetto di sistema.

    Nota: a questo punto vi raccomandiamo caldamente di creare un file MEM. SAV (vedi paragrafo sul MEM.SAV in questo capitolo). MEM.SAV riserva un certo numero di settori sul dischetto di sistema (e sugli altri dischetti) per memorizzare il programma RAM residente mentre usate la funzione DOS cioè, il file DUP.SYS (vedere il capitolo che spiega la funzione DUP.SYS).

11. Togliete il dischetto di sistema appena creato dal Drive 2 e inseritelo nel Drive 1. Riponete il dischetto principale nel contenitore di carta e conservatelo con cura (vedere capitolo sul come conservare i dischetti).

12. Battete **N** e premete **RETURN** per creare un file MEM.SAV sul vostro dischetto di sistema.

13. Quando appare il messaggio TYPE Y TO CREATE MEM.SAV (battete Y per creare un File MEM.SAV) battete **Y** e premete **RETURN**.

14. Quando appare il messaggio SELECT ITEM OR RETURN FOR MENU, sul vostro dischetto di sistema vi è ora un file MEM.SAV.

Siccome il DOS II occupa 10 Kbytes della memoria dati disponibile, non è pratico mettere il DOS su ogni dischetto, caricate i dischetti nell'ordine seguente:

1. caricate il dischetto di sistema

2. sostituite il dischetto di sistema con il dischetto che intendete usare.

RICORDATE: se, per qualsiasi ragione, spegnete il sistema, dovrete rimuovere ogni dischetto che non abbia il DOS II su di esso e inserire il dischetto di sistema prima di poter riutilizzare il computer.

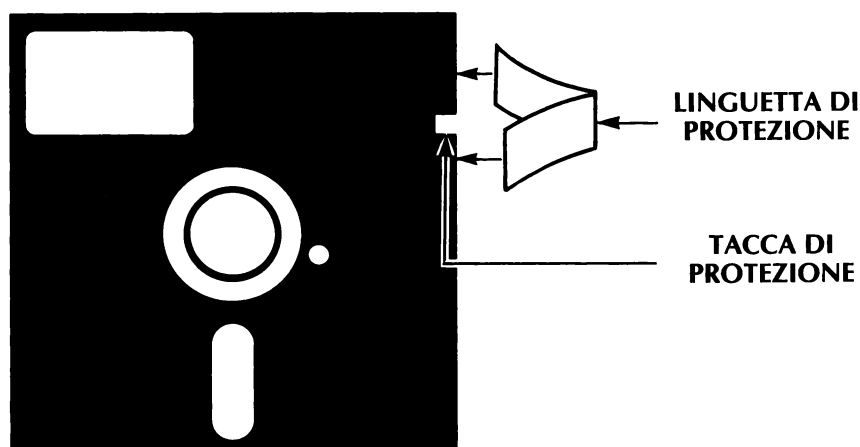



Figura 2-2 Protezione di un dischetto.

COME PROTEGGERE I VOSTRI DISCHETTI

La protezione è un metodo semplice per salvaguardare i vostri dischetti dalla scrittura accidentale che cancellerebbe da un dischetto preziose informazioni che non volete perdere.

Noterete che il dischetto principale non ha la tacca sulla sinistra cosicché è impossibile scrivere file su di esso, dunque è già protetto. I dischetti blank preformattati hanno, invece, la tacca sulla sinistra, il che rende possibile scrivere sul dischetto.

Normalmente non proteggerete i vostri dischetti poiché ciò disabilita la funzione MEM.SAV, impedendovi dal salvare il programma residente in RAM quando necessario. Per questa ragione sarà opportuno registrare i file che volete salvare su un altro dischetto, che potete proteggere.

COME PROTEGGERE I DISCHETTI PREZIOSI

Un foglio di etichette per l'identificazione dei file e un secondo foglio di piccole linguette adesive di protezione sono fornite in ogni confezione di dischetti ATARI. La protezione si attua semplicemente ponendo una linguetta di protezione sull'orlo del dischetto in modo che copra la tacca di protezione (vedere figura 2-2).

Se state usando una unità doppia a dischi ATARI 815, potete proteggere i vostri dischetti usando l'interruttore di protezione (WRIT PROT) sull'unità a dischi. Quando questo interruttore è acceso (colore rosso) vi viene impedito di aggiungere, cambiare e cancellare file. Quando però non avete più bisogno della condizione di protezione, RICORDATE di spegnere l'interruttore WRIT PROT.

Se tentate di scrivere su un dischetto protetto, verrà visualizzato il messaggio: ERROR-144 sullo schermo.

COME ETICHETTARE I DISCHETTI

Usate le etichette autoadesive per identificare i dati su ogni dischetto. Scrivete sull'etichetta prima di porla sul dischetto per evitare che quest'ultimo venga danneggiato, quindi attaccatela sulla parte superiore destra dell'involucro.

REGISTRAZIONE SU DISCHETTI A DOPPIA E SINGOLA DENSITÀ

La principale differenza fra l'unità a dischi ATARI 810 e l'unità doppia a dischi ATARI 815 stà nel modo in cui i dati sono codificati per la memorizzazione sui dischetti.

ATARI 810	ATARI 815
Le informazioni vengono trasferite all'unità a dischi in blocchi di 128 bytes (densità singola) e ogni blocco di 128 bytes riempie un settore del dischetto.	Le informazioni vengono trasferite all'unità a dischi in blocchi di 256 bytes (densità doppia) e ogni blocco di 256 bytes riempie un settore del dischetto.

Il DOS II è disponibile in due versioni:

- 2.0S per registrare i dati in densità singola sull'unità a dischi ATARI 810
- 2.0D per registrare i dati in doppia densità sulla unità a dischi doppia ATARI 815.

QUALI DISCHETTI USARE?

Per svolgere con successo le operazioni con dischi dovete avere la versione corretta del DOS II per la vostra (le vostre) unità a dischi e i giusti dischetti blank per la memorizzazione dei dati. La tabella in figura 2-3 mostra quali dischetti potete usare in relazione alla (alle) unità a dischi che possedete:

Unità a dischi ATARI 810

Unità doppia a dischi ATARI 815

Combinazione delle unità a dischi ATARI 810 e ATARI 815

UNITÀ A DISCO	CX8104 DISCHETTO PRINCIPALE ATARI 810	CX8111 DISCHETTO FORMATTATO ATARI 810	CX8100 DISCHETTO BIANCO ATARI 810	CX8202 DISCHETTO BIANCO ATARI 810/815	CX8201 DISCHETTO PRINCIPALE ATARI 815
ATARI 810	X	X	X	X	
ATARI 815				X	X

Figura 2-3 Dischetti giusti per la vostra unità a dischi

La figura 2-3 mostra che potete usare solo il dischetto principale II ATARI 810 (CX8104) (versione a densità singola del DOS II) con l'unità a dischi ATARI 810. Potete scegliere fra i dischetti in bianco ATARI 810 (CX8100), i dischetti in bianco ATARI 810/815 (CX8202) e i dischetti formattati II ATARI 810 (CX8111) per la memorizzazione di dati e programmi.

Se possedete una o più unità doppia ATARI 815 usate il dischetto principale ATARI 815 (CX8201) (versione a doppia densità del DOS II) e il dischetto in bianco ATARI 810/815 (CX8202) che viene fornito con l'unità a dischi. Siccome i dischetti CX8202 sono completamente in bianco, dovrete formattarli prima di poterli usare per duplicare il dischetto principale (vedere capitolo su come formattare un disco).

Se avete sia l'ATARI 810 che l'ATARI 815 collegati al vostro personal computer ATARI usate il dischetto principale ATARI 815 (CX8201) e inseritelo nel drive 1 della doppia unità a dischi ATARI 815. ASSICURATEVI CHE OGNI DRIVE abbia un numero di codice differente. Assegnate all'unità a dischi ATARI 810 il numero 3.

COME CONSERVARE I DISCHETTI

I vostri dischetti, essendo flessibili, sono soggetti a danni; i suggerimenti che seguono vi aiuteranno a mantenere i vostri dischetti in buone condizioni:

- Tenete SEMPRE i dischetti nel loro contenitore di carta quando non li usate.
- Conservateli verticalmente come fareste con i dischi di musica; non impilateli uno sull'altro.
- Conservate i dischetti ad ALMENO 30 cm dal vostro televisore o da qualsiasi altra sorgente di campi magnetici.
- Tenete i dischetti lontano da ogni fonte diretta di calore.

I vostri dischetti sono una parte importante del vostro sistema ATARI e, con l'opportuna cura, vi daranno molte ore di sicuro uso e divertimento.

3

USO DEL DOS II

Questo capitolo e i seguenti vi insegnano a lavorare con i file.

IDENTIFICAZIONE DEI FILE SU DISCO

I file sono classificati in due tipi:

PROGRAMMI: sono set di istruzioni che dicono al computer come eseguire un determinato compito.

DATI: contengono, generalmente, le informazioni usate da un programma, ma non le istruzioni. Per esempio, un file di dati permanenti può contenere nomi e indirizzi e può venire aggiornato in qualsiasi momento.

Proprio come si chiama una persona per nome, così dovete chiamare ogni file per nome per potervi accedere. Il nome del file (filename) sul dischetto è parte delle informazioni (file specifications o, abbreviato, Filespec) che riguardano il file che noi indicheremo con il termine identificatore. Queste informazioni (vedere figura 3-1) hanno sei elementi chiave. Se chiamate un file col nome sbagliato, proprio come una persona, non risponderà; vedrete invece un: ERROR-170 apparire sullo schermo del televisore.

Le regole per i nomi dei file sono:

- La lunghezza massima di un nome è di otto caratteri.
- I soli caratteri consentiti sono le lettere dalla A alla Z e le cifre da 0 a 9.
- Il primo carattere è SEMPRE un carattere alfabetico.
- I caratteri * e ? NON POSSONO entrare nel nome di un file (vedere il capitolo sui caratteri aggiuntivi per spiegazioni).
- I nomi DOS.SYS, DUP.SYS, AUTORUN.SYS e MEM.SAV sono riservati al DOS II.

ESTENSIONI DEI NOMI E LORO USO

Potete aggiungere un'estensione di tre caratteri al nome del file, per indicare il tipo di informazioni nel file. Potete usare qualsiasi combinazione valida di lettere e numeri, per esempio:

SYS file di sistema
BAS programma BASIC
DAT file di dati
MUS file: ATARI Music composer (compositore di musica ATARI)
ASM file di linguaggio Assembler
OBJ file di caricamento binario
SRC file sorgente (source file)
LST file creato dall'istruzione LIST
SVE file creato dal comando SAVE

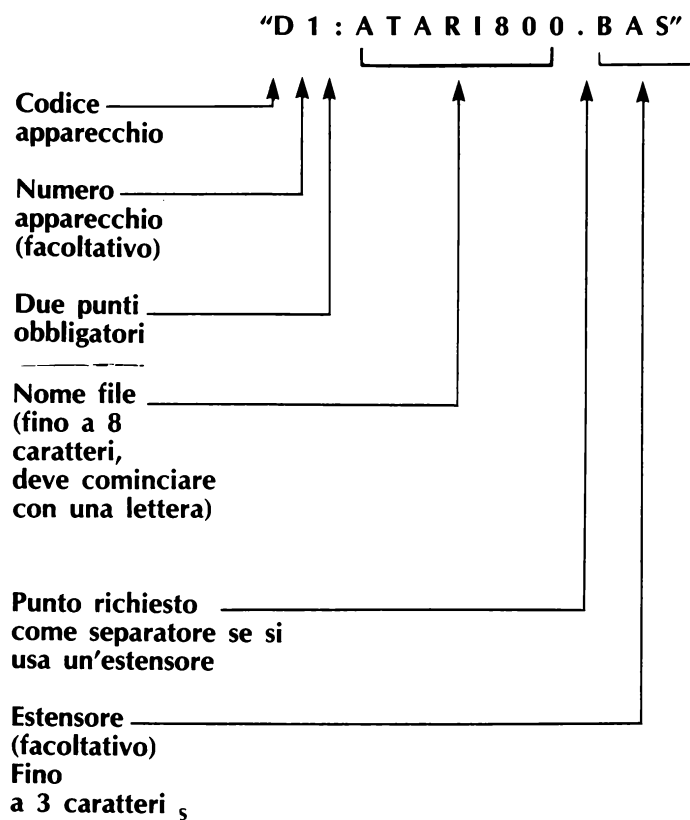



Figura 3-1 Struttura di un Filespec

Se cercate di usare un'estensione che abbia più di tre caratteri, il DOS II ignorerà i caratteri in più. Gli esempi che seguono illustrano nomi di file sia corretti che scorretti con una spiegazione di cosa rende il nome scorretto.

CASHFLOW	Nome corretto.
ATARI.BAS	Nome corretto.
3ATARI.DAT	Nome scorretto, il primo carattere non è una lettera dell'alfabeto.
ATARI22.ASM	Nome corretto.
ATARI#	Nome scorretto.
A1234567.BA2	Nome corretto.
B ATARI.LIST	Nome scorretto, non sono ammessi spazi.
DOS.SYS	Nome scorretto, riservato al DOS.
DOSSYS	Nome corretto.
TEST1.123	Nome corretto.
ATARI.BASIC	Nome corretto. Notate che il DOS ignora le ultime due lettere dell'estensore.

Figura 3-2 Esempio di Filename corretti e scorretti

CARATTERI JOLLY (WILD CARDS)

Il DOS ATARI riconosce due caratteri jolly che potete inserire **al posto** di altri caratteri nel nome di un file. I caratteri sostitutivi sono rappresentati dal punto interrogativo (?) e dall'asterisco (*).

Il punto interrogativo sostituisce un **singolo** carattere, l'asterisco sostituisce qualsiasi combinazione valida di caratteri e, dunque, è più flessibile. Gli esempi che seguono illustrano l'uso dell'asterisco e del punto interrogativo.

Esempi:

*.BAS	Listerà tutti i programmi sul dischetto nel drive 1 che terminano per .BAS.
D2:*.*	Listerà tutti i programmi sul dischetto nel drive 2.
PRO*.BAS	Listerà tutti i programmi sul dischetto nel drive 1 che cominciano con PRO e abbiano BAS come estensione.
TEST??	Listerà tutti i programmi sul dischetto nel drive 1 che comincino con TEST ed abbiano qualsiasi combinazione di lettere o numeri per gli ultimi due caratteri.

La figura 3-3 riassume tutte le operazioni del Menu DOS e mostra quali di queste ammettono caratteri jolly nei loro parametri.

OPZIONE DEL MENU DOS	CARATTERE AGGIUNTIVO
A. Disk Directory	Si
B. Run Cartridge	No
C. Copy File	Si
D. Delete File	Si
E. Rename File	Si
F. Lock File	Si
G. Unlock File	Si
H. Write DOS File	No
I. Format Disk	No
J. Duplicate Disk	No
K. Binary Save	No
L. Binary Load	No
M. Run at Address	No
N. Create MEM.SAV	No
O. Duplicate File	Si

Fig. 3-3 Opzioni del Menu che possono o meno usare caratteri sostitutivi (wild cards)

ERRORI D'"INIZIO" (BOOT ERRORS)

Quando accendete il vostro sistema, si possono avere errori d'"inizio" (boot errors) per le seguenti ragioni (vedere figura 3-4):

1. Il dischetto inserito non ha il DOS su di esso.
2. Il dischetto è stato inserito male.
3. Il dischetto è stato danneggiato, usatene un altro.

Nota: per un dischetto formattato II ATARI 810 (CX8111), questo dovrebbe essere fatto SOLO COME ULTIMA RISORSA per non perdere la maggiore velocità del dischetto preformattato.

6. Se anche questo fallisce, il dischetto deve essere scartato.

COME SALVARE, CARICARE ED ESEGUIRE PROGRAMMI

Dopo aver creato il vostro dischetto di sistema e (se necessario) aver formattato un dischetto blank, siete pronti per scrivere i vostri programmi. Quando spegnete il vostro computer, perdete ogni programma che si trova in memoria. Con un'unità a dischi ATARI, avete la possibilità di memorizzare e richiamare i vostri programmi, senza doverli ribattere. Potete usare le seguenti istruzioni BASIC per salvare e richiamare i programmi. Abbiamo incluso un programma tipo da scrivere sul computer e una procedura che, passo passo, vi mostri come salvarlo su un dischetto e ricaricarlo nel computer. Usate il primo set di istruzioni se avete una unità a dischi ATARI 810 e il secondo set se avete una unità doppia a dischi ATARI 815.

Se avete un'unità a dischi ATARI 810:

1. Accendete l'unità a dischi.
2. Inserite il dischetto di sistema nel Drive 1.
3. Accendete la console del computer e il televisore.
4. Quando la spia BUSY si spegne, togliete il dischetto di sistema ed inseritene uno che sia stato formattato.
5. Battete il programma in figura 3-5.
6. Battete **SAVE"D:INTEREST.SAV" RETURN**.
7. Quando la spia BUSY si spegne e appare il messaggio READY, il programma che avete battuto è stato salvato con successo sul dischetto.
8. Battete **NEW RETURN** per cancellare il programma dalla RAM.

   Ora, per ricaricare il programma in memoria:

9. Battete **LOAD "D:INTEREST.SAV" RETURN**.
10. Quando appare il messaggio READY potete eseguire il programma battendo **RUN RETURN**.
11. Potete anche caricare ed eseguire il programma battendo **RUN "D:INTEREST-.SAV" RETURN**.

    NOTA: non cancellate il programma "Interest" dal dischetto, lo usurete ancora nel capitolo 5.

```
100 REM *** INTERESSI
110 PRINT "SE BATTETE L' AMMONTARE DEL CAPITALE"
120 PRINT "INIZIALE E IL TASSO DI INTERESSE ANNUO";
130 PRINT "IO MOSTRERO' COME I VOSTRI SOLDI AUMEN"
140 PRINT "TANO ANNO DOPO ANNO, PER FERMARMI PRE"
150 PRINT "MI IL TASTO BREAK."
155 PRINT
160 PRINT "PRINCIPALE"
165 INPUT P
170 PRINT "TASSO DI INTERESSE"
175 INPUT R
180 LET N=1
190 PRINT
200 LET A=P*(1+R/100)^N
210 PRINT "ANNO=";N
220 PRINT "AMMONTO=";A
230 LET N=N+1
240 GOTO 190
```

READY

Figura 3-5 Programma tipo di interesse

Se avete un'unità doppia a dischi ATARI 815:

1. Accendete l'unità a dischi.
2. Inserite il dischetto di sistema nel drive 1 e un dischetto formattato nel drive 2.
3. Accendete il computer e il televisore.
4. Battete il programma in figura 3-5.
5. Battete **SAVE "D2:INTEREST.SAV" RETURN**.
6. Quando la spia BUSY si spegne e appare il messaggio READY sullo schermo, il programma è stato salvato con successo sul dischetto nel drive 2.
7. Battete **NEW RETURN** per cancellare il programma della RAM.

   Ora potete caricare il programma che avete salvato.
8. Battete **LOAD "D2:INTEREST.SAV" RETURN**.
9. Il programma è pronto per essere eseguito. Battete **RUN RETURN**.
10. Potete anche caricare ed eseguire il vostro programma battendo **RUN "D:INTEREST.SAV" RETURN**.

    NOTA: non cancellate questo programma dal dischetto, lo userete ancora nel capitolo 5.

4

SELEZIONE DI UN'OPZIONE DEL MENU DOS

Per selezionare un'opzione del Menu DOS:

1. Battete **DOS** e premete **RETURN**.
2. Il Menu apparirà sullo schermo listando le 15 opzioni disponibili. Vedere figura 1-4 "Il Menu DOS II".
3. Fate la vostra scelta e premete **RETURN**.
4. Apparirà un messaggio che elenca i parametri che dovete fornire prima che il DOS possa eseguire la funzione scelta. I parametri sono informazioni aggiuntive (a volte facoltative) che specificano come l'istruzione deve operare.
5. Il messaggio SELECT ITEM OR RETURN FOR MENU appare ogniqualvolta il sistema completa una funzione. Se decidete di selezionare un'altra opzione, battete la lettera corrispondente e premete **RETURN**. La metà inferiore dello schermo eseguirà uno scroll verso l'alto per permettere la visualizzazione del successivo messaggio/i. Se battete **RETURN** lo schermo si cancellerà e riapparirà il Menu DOS.

A. DISK DIRECTORY

La Disk Directory (Elenco dei file sui dischi) contiene una lista di tutti i file su dischetto. Questa opzione visualizza i filenames, le estensioni (se ve ne sono) e il numero dei settori che questo file occupa; visualizza inoltre una lista parziale o completa a seconda dei parametri inseriti. I caratteri jolly (wild cards) possono essere usati nei parametri.

Battete **A** e premete **RETURN** al messaggio SELECT ITEM OR RETURN FOR MENU. Lo schermo visualizza immediatamente il messaggio:

DIRECTORY—SEARCH SPEC, LIST FILE
(DIRECTORY—SPECIFICA DA CERCARE, LISTA DEL FILE)

Se premete ancora **RETURN** dopo questo messaggio, vedrete una lista di tutti i nomi di file sul dischetto, le dimensioni (in settori) di ogni file e il numero di settori liberi che rimangono sul dischetto. L'esempio seguente mostra l'elenco dei file del vostro dischetto di sistema per il DOS II:

SELECT ITEM OR RETURN FOR MENU
A **RETURN**
DIRECTORY—SEARCH SPEC, LIST FILE
RETURN
RETURN — Lista sullo schermo tutti i filename dal dischetto nel drive 1

DOS.SYS 039 (unità a dischi ATARI 810)
DOS.SYS 019 (unità doppia a dischi ATARI 815)
DUP.SYS 042 (unità a dischi ATARI 810)
DUP.SYS 021 (unità doppia a dischi ATARI 815)
MEM.SAV 045 (unità a dischi ATARI 810)
MEM.SAV 022 (unità doppia a dischi ATARI 815)
581 FREE SECTORS (581 settori liberi; unità a dischi ATARI 810)
626 FREE SECTORS (626 settori liberi; unità doppia a dischi ATARI 815).

SELECT ITEM OR RETURN FOR MENU

PARAMETRI PER L'OPZIONE DISK DIRECTORY

Come potete vedere dal messaggio per la Disk Directory questa istruzione ha due parametri:

SEARCH SPEC e LIST FILE
IDENTIFICATORE e LISTATO

Se non indicate un preciso identificatore di un file (filespec) in questo parametro, il DOS porrà il valore di: D1:*.*,E: per questi due parametri. Il primo parametro D1:*.* dice al DOS che volete una lista dei nomi e delle dimensioni di tutti i file presenti sul dischetto che si trova nel Drive 1. Potete scegliere di ricercare un singolo file, diversi file o tutti i file sul dischetto designato. Se non specificate l'unità a dischi, il DOS II assumerà che vogliate vedere i file sul dischetto nel Drive 1.

Il secondo parametro, E:, dice al DOS che volete visualizzate sul video queste informazioni. Dunque, se non specificate alcun parametro e premete semplicemente **RETURN**, il DOS listerà sullo schermo i nomi e le dimensioni dei file memorizzati sul dischetto nel Drive 1.

Se avete una stampante ATARI, potete stampare una copia permanente dell'elenco usando **P:** come secondo parametro. Nell'esempio che segue, i dati sono stampati per un solo file, DOS.SYS.

1. Battete **A** e premete **RETURN**.
2. Dopo il messaggio battete **DOS.SYS,P:** e premete **RETURN**.
3. Se avete una stampante e questa è accesa, una lista parziale per il Drive 1 sarà stampata anziché essere visualizzata sullo schermo.

Sullo schermo o sulla stampante vedrete:

DOS.SYS 039 (per la densità singola)
DOS.SYS 019 (per la densità doppia)

Se non avete una stampante (o se è spenta) vedrete apparire sullo schermo: ERROR-138. Ogniqualvolta l'opzione DISK DIRECTORY completa una funzione, viene visualizzato il messaggio: SELECT ITEM OR RETURN FOR MENU. La figura 4-1 mostra diversi modi di utilizzare questa opzione.

Nota: quando vengono mostrati i nomi dei file, i nomi e le loro estensioni sono separati da uno spazio. Tuttavia, quando volete accedere ad un file, DOVETE usare un punto fra il nome e la sua estensione.

Esempio 1

SELECT ITEM OR RETURN FOR MENU	
A **RETURN**	
DIRECTORY—SEARCH SPEC, LIST FILE?	
***.*SYS RETURN**	Lista sullo schermo dei file con l'estensione .SYS che si trovano sul dischetto nel Drive 1.
SELECT ITEM OR RETURN FOR MENU	

Esempio 2:

SELECT ITEM OR RETURN FOR MENU	
A **RETURN**	
DIRECTORY—SEARCH SPEC, LIST FILE?	
D2:,P: RETURN	Lista sulla stampante di tutti i file sul dischetto nel Drive 2.
SELECT ITEM OR RETURN FOR MENU	

Esempio 3:

SELECT ITEM OR RETURN FOR MENU	
A **RETURN**	
DIRECTORY—SEARCH SPEC, LIST FILE?	
EO?.* RETURN	Lista di tutti i file col nome di 3 lettere che inizi con EO e si trovino sul dischetto nel Drive 1.
SELECT ITEM OR RETURN FOR MENU	

Figura 4-1 Uso dell'opzione Disk Directory

B. RUN CARTRIDGE

Quando selezionate B il DOS II passa il controllo del sistema al cartridge inserito nel vostro Personal Computer ATARI. Se è inserito il Cartridge BASIC, lo schermo visualizza il messaggio READY, se è inserito il Cartridge Assembler Editor lo schermo visualizza il messaggio EDIT. Se non è inserito alcun cartridge apparirà sullo schermo il messaggio NO CARTRIDGE.

Esempio:

SELECT ITEM OR RETURN FOR MENU
B RETURN

Se esiste il file MEM.SAV sul dischetto nel Drive 1 il programma in linguaggio BASIC o Assembly sarà salvato sul dischetto quando battete **DOS RETURN** e ricaricato in RAM quando il controllo ritorna al cartridge (B.RUN CARTRIDGE). Si assume che il dischetto nel Drive 1 sia lo stesso che c'era prima di chiamare il DOS e che non abbiate invalidato il MEM.SAV con l'uso di COPY FILE, DUPLICATE FILE o DUPLICATE DISK. Apparirà un messaggio per ricordarvi che MEM.SAV può essere invalidato se usate una di queste istruzioni (vedi il capitolo sul MEM.SAV).

Se non vi era un file MEM.SAV sul dischetto di sistema (nel Drive 1) quando avete chiamato il DOS, scoprirete che il programma in BASIC o in linguaggio Assembly che c'era prima di chiamare il DOS è ora sparito. Il vostro programma non può essere richiamato, a meno che non lo abbiate salvato su un dischetto prima di chiamare il DOS. Questa perdita avviene quando usate DOS II perché spartite l'area di memoria riservata al programma con le routine di utility su disco (disk utility package) memorizzate nel file DUP.SYS. La spartizione della RAM col DUP-.SYS incrementa l'ammontare di RAM disponibile all'utente rispetto all'uso del DOS I.

C. COPY FILE

Usate questa opzione quando avete due o più unità a dischi e volete copiare un file da un dischetto in un drive su un altro dischetto in un secondo drive. Ci sono due parametri associati all'istruzione COPY FILE: FROM (DA) e TO (A). Il primo parametro FROM è generalmente un identificatore di file (filespec) che può o meno contenere caratteri jolly (wild cards). L'uso di caratteri jolly nel primo parametro vi fornisce un ottimo sistema per copiare un gruppo di file da un drive a un altro (vedere esempio 6). L'opzione /A può essere usata col secondo parametro per permettere l'unione di due file complementari. Il secondo parametro è generalmente un'identificatore di file (filespec) ma può essere anche una periferica del tipo E: (video), P: (stampante), D: (unità a dischi) (vedere esempi 3, 5 e 6 in fig. 4-2).

COPY FILE può anche essere usata per creare una copia di un determinato file sullo stesso dischetto ma con una diversa estensione o anche con un nome completamente diverso. Se il file che state copiando con un nuovo nome è fatto di diversi file che sono stati uniti (un file "composto") la nuova versione del file sarà "compressa" cioè occuperà meno settori del file originale.

Nota: il tentativo di copiare il file DOS.SYS genererà un messaggio d'errore. L'unico modo per "scrivere" un file DOS.SYS è di usare l'opzione H. WRITE DOS.SYS (scrivi il DOS. SYS).

Se tentate di copiare un file, come descritto più sopra, quando un file MEM.SAV si trova sul vostro dischetto di sistema otterrete un nuovo messaggio. Otterrete il nuovo messaggio dopo aver battuto il numero del drive sorgente (da dove proviene l'informazione) e il numero del drive destinazione (dove il dato è inviato). Questo messaggio, TYPE Y FOR OK TO USE PROGRAM AREA CAUTION: A Y WILL INVALIDATES MEM.SAV (batti Y per autorizzare l'uso dell'area dei programmi, attenzione una Y invaliderà MEM.SAV) appare per ricordarvi che il DOS II può usare l'area di programma per accelerare il processo di copiatura del file. Una "Y" notifica al DOS II che non vi importa dell'area di programma e nemmeno del MEM.SAV e il MEM.SAV sarà invalidato. Una **N** dirà al DOS II che non può porre niente nell'area di programma; può usare solo un buffer interno molto più piccolo per spostare il file. In altre parole, il vostro file sarà comunque copiato ma occorrerà più tempo.

Potete usare questa opzione anche per copiare il listato del file sullo schermo (**E:**) o sulla stampante (**P:**).

Avvertenza 1: non unite file BASIC "tokenizzati" cioè file memorizzati con l'istruzione SAVE. Tuttavia potete unire due file BASIC memorizzati con una istruzione LIST o due file binari creati con il cartridge Assembler Editor o con il DOS II (i file tokenizzati e non tokenizzati sono spiegati nel capitolo 5).

Avvertenza 2: ricordate che in una operazione di fusione (merge operation), i file memorizzati con l'istruzione LIST che hanno uguali numeri di linea possono interferire fra di loro.

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
D1:DOSEX. BAS, D2:DOSEX.BAS **RETURN** — Copia DOSEX.BAS da D1 a D2.

SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
D1:DOSEX.BAS,D1:DOSEX.BAK — Crea una copia del file sullo stesso dischetto.

SELECT ITEM OR RETURN FOR MENU

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
D1:DOSEX.LST,E: **RETURN** — Visualizza sullo schermo il listato del programma.

SELECT ITEM OR RETURN FOR MENU

Esempio 4:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
E:,D1:TEMP.DAT **RETURN** — Copia ogni dato in un file chiamato TEMP.DAT. Battete sullo schermo i dati che volete memorizzare nel file TEMP.DAT.

PETER **RETURN**
BILL **RETURN**
RAY **RETURN**
STEVE **RETURN**
CTRL 3 — Termina l'inserimento di dati.
SELECT ITEM OR RETURN FOR MENU

Esempio 5:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
D1:DISEX.LST,P.: **RETURN** — Trasferisce su stampante il listato del programma DISEX.LST.

SELECT ITEM OR RETURN FOR MENU

Esempio 6:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
.,D2: **RETURN**

Copia tutti i file da D1 a D2 tranne quelli che hanno estensione .SYS.

SELECT ITEM OR RETURN FOR MENU

Esempio 7:

SELECT ITEM OR RETURN FOR MENU
C **RETURN**
COPY—FROM, TO?
D1:PROG2,PROG1/A **RETURN**

Unisce il file PROG2 in D1 al file PROG1.

SELECT ITEM OR RETURN FOR MENU

Figura 4-2 Uso dell'opzione Copy File

D. DELETE FILE

Questa opzione vi consente di cancellare uno o più file da un dischetto ed il loro nome dalla lista dei file (Disk Directory).

Possono essere usati caratteri sostitutivi nei nomi dei file.

Nota: DOS II non vi permette di cancellare alcun file da un dischetto formattato dal DOS I; per cancellare tali file dovete usare il DOS I.

Il messaggio di verifica vi dà la possibilità di cambiare idea sulla cancellazione del file. Aggiungendo l'opzione /N (non è richiesta verifica) alla specifica del file (filespec) in entrata, il DOS II eliminerà questa verifica (vedi esempio 3 in fig. 4-3).

Potete anche cancellare tutti i file dal dischetto pur lasciando formattato il dischetto. L'esempio 4 illustra come cancellare tutti i file esistenti sul dischetto nel Drive 1. Notate che nell'esempio è stata usata l'opzione /N per evitare la verifica ad ogni file del dischetto. Se tentate di cancellare un file bloccato (locked) lo schermo mostrerà: ERROR-167 (File Bloccato).

Se avete acquistato i dischetti formattati II ATARI 810, questo è un modo eccellente per cancellare il dischetto senza distruggere il formattamento disponibile su questi dischetti.

E. RENAME FILE

Questa opzione vi permette di cambiare il nome a uno o più file. Vi sono due parametri, OLD NAME (nome vecchio) e NEW (nuovo) per questa opzione. Il parametro OLD NAME è l'identificatore di un file (filespec). Se non specificate il numero della periferica, il computer assumerà che si tratti di D1:. Il parametro

NEW si riferisce semplicemente al nuovo nome; la periferica è automaticamente specificata nei parametri di OLD NAME. Se ci sono caratteri scorretti nel parametro NEW, il nuovo nome del file consisterà nei caratteri fino al carattere scorretto escluso. Potete usare caratteri jolly (wild cards) sia nel primo che nel secondo parametro (vedi esempio 2 in fig. 4-4).

Avvertenza: non cambiate nome ai file su un dischetto DOS II usando DOS I.

In generale non dovreste mai usare DOS I coi dischetti DOS II. Se tentate di cambiare nome a un file su un dischetto protetto apparirà: ERROR-144 (la periferica non esegue). Se cercate di cambiare nome a un file che non è sul dischetto apparirà: ERROR-170 (file non trovato); se lo schermo mostra ERROR-167 significa che avete tentato di cambiare nome a un file bloccato (vedere F. LOCK FILE).

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
D **RETURN**
DELETE FILESPEC
D2:REM*.BAS **RETURN** — Tutti i file che iniziano con REM e hanno .BAS come estensore.
TYPE "Y" TO DELETE... — Messaggio di verifica.
REM1.BAS? — Cancella REM1.BAS
Y **RETURN**
REMBAA. BAS — Cancella REMBAA.BAS.
Y **RETURN**
SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
D **RETURN**
DELETE FILESPEC
D: TEMP.DAT **RETURN** — Un singolo file. Messaggio di verifica.
TYPE "Y" TO DELETE...
TEMP.DAT
N **RETURN** — Se viene battuta Y il file sarà cancellato.

SELECT ITEM OR RETURN FOR MENU

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
D **RETURN**
DELETE FILESPEC
DOXEX.BAS/N **RETURN** — Il file sarà cancellato senza la richiesta di verifica.

SELECT ITEM OR RETURN FOR MENU

Esempio 4:

SELECT ITEM OR RETURN FOR MENU
D **RETURN**
DELETE FILESPEC
***.*/N** **RETURN** — Cancella tutti i file dal dischetto nel Drive 1.

SELECT ITEM OR RETURN FOR MENU

Figura 4-3 Uso dell'opzione Delete File

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
E **RETURN**
RENAME, GIVE OLD NAME, NEW
D2: TEMP.DAT,NAMES.DAT **RETURN** — Cambia il file del Drive 2 da TEMP.DAT a NAMES.DAT.

SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
E **RETURN**
RENAME, GIVE OLD NAME, NEW
.8KB,.BAS **RETURN** — Tutti i file con estensione 8KB hanno cambiato le loro estensioni in .BAS.

SELECT ITEM OR RETURN FOR MENU

Figura 4-4 Uso dell'opzione Rename File

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
F **RETURN**
WHAT FILE TO LOCK?
DOS.SYS **RETURN** — Blocca il file DOS.SYS
SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
F **RETURN**
WHAT FILE TO LOCK?
D1:*.BAS **RETURN** — Blocca tutti i file su D1 con l'estensione .BAS.

SELECT ITEM OR RETURN FOR MENU

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
F **RETURN**
WHAT FILE TO LOCK?
T*.* **RETURN** — Blocca tutti i file D1 che iniziano con T.

SELECT ITEM OR RETURN FOR MENU

Esempio 4:

SELECT ITEM OR RETURN FOR MENU
F **RETURN**
WHAT FILE TO LOCK?
. **RETURN** — Blocca tutti i file in D1.
SELECT ITEM OR RETURN FOR MENU

Figura 4-5 Uso dell'opzione Lock File

F. LOCK FILE

Usate questa opzione per proteggere un singolo file. Un file bloccato non può essere modificato, unito ad altri, cambiato di nome o cancellato. Apparirà un: ERROR-167 tentando di scrivere su un file bloccato. Potete usare caratteri jolly per bloccarne diversi contemporaneamente.

Un file bloccato apparirà nella lista dei file (Disk Directory) preceduto da un asterisco (*). NON confondete questo asterisco con un carattere jolly (wild cards).

Avvertenza: se bloccate qualche file e quindi formattate il dischetto, i file bloccati SARANNO COMUNQUE CANCELLATI. In altre parole, l'opzione FORMAT DISKETTE ignora l'istruzione LOCK FILE.

G. UNLOCK FILE

Usate questa opzione per sbloccare i file che avete bloccato in precedenza usando l'opzione F. Quando questa opzione viene eseguita, l'asterisco che appariva prima del nome nella lista dei file (disk directory) a indicare che quel file era bloccato, non apparirà più sullo schermo quando userete ancora l'istruzione DISK DIRECTORY (Menu DOS, opzione A). Possono essere usati caratteri jolly (wild cards) nei nomi dei file.

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
G **RETURN**
WHAT FILE TO UNLOCK?
DOSEX.BAS **RETURN** — Sblocca il file DOSEX.BAS sul Drive 1.

SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
G **RETURN**
WHAT FILE TO UNLOCK?
T*.* **RETURN** — Sblocca i file che iniziano con T nel Drive 1.

SELECT ITEM OR RETURN FOR MENU

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
G **RETURN**
WHAT FILE TO UNLOCK?
PROB?.DAT **RETURN**

SELECT ITEM OR RETURN FOR MENU

Sblocca tutti i file con nomi di 5 lettere che iniziano con PROB. e hanno .DAT come estensione.

Figura 4-6 Uso dell'opzione Unlock File

H. WRITE DOS FILE

Per riscrivere il DOS II (composto dal file DOS.SYS e DUP.SYS) su un dischetto dovete aver formattato un dischetto in precedenza (vedere I. FORMAT DISK) o anche usare un dischetto formattato II ATARI 810 (CX8111) (Il dischetto sul quale DOS II dovrà essere riscritto può essere inserito in un drive a scelta).

Nota: il DOS II non vi permetterà di riscrivere il nuovo DOS II su un dischetto formattato dal DOS I; così, il DOS I dovrà essere usato solo per riscrivere il file DOS (WRITE DOS FILE) su un dischetto formattato dal DOS I. Inoltre, come potete supporre, il DOS I non vi permetterà di scrivere una copia del DOS I su un dischetto formattato dal DOS II.

Avvertenza: non dovete mai riscrivere il file DOS su un dischetto DOS II. Quando lavorate con qualche dischetto formattato dal DOS I e con altri formattati dal DOS II, usate DOS II per premunirvi da errori che potrebbero danneggiare i vostri preziosi dischetti.

Non appena i file DOS sono stati scritti (vedere figura 4-7) lo schermo è cancellato e vengono visualizzati sia il menu che il messaggio SELECT ITEM OR RETURN FOR MENU.

Se tentate di scrivere un nuovo file DOS su un dischetto che è stato protetto, otterrete un ERROR-144. Otterrete inoltre un errore tentando di riscrivere un nuovo file DOS su un dischetto inserito in un'unità a dischi ATARI 815 che abbia la spia WRIT PROT accesa.

SELECT ITEM OR RETURN FOR MENU
H **RETURN**
DRIVE TO WRITE DOS FILE TO?
1 **RETURN**
TYPE "Y" TO WRITE DOS TO DRIVE 1.
Y **RETURN**
WRITING NEW DOS FILES

SELECT ITEM OR RETURN FOR MENU

Figura 4-7 Uso dell'opzione Write DOS File

I. FORMAT DISKETTE

Questa opzione è usata per formattare un dischetto. Il dischetto può essere vuoto o contenere file che non vi interessano più. Il formattamento scrive sul dischetto le informazioni che permettono ai dati di esservi memorizzati e ripresi. Questa operazione dura circa 2 minuti sull'unità a dischi ATARI 810 e 2,5 minuti sull'unità a dischi doppia ATARI 815.

Gli esempi in figura 4-8 illustrano il formattamento del DRIVE 1; tuttavia, potete specificare qualsiasi drive alla richiesta WHICH DRIVE TO FORMAT? (quale Drive per formattare?). Non è possibile formattare un dischetto contenente settori danneggiati. Lo schermo mostrerà un ERROR-173 (settori danneggiati all'operazione di formattamento) e il DOS II si rifiuterà di formattare il dischetto. Se il DOS II riceve dal drive il messaggio che il dischetto contiene settori danneggiati, esso cercherà di formattarlo ancora due volte. Se ciò accade, possono trascorrere fino a 15 minuti nel tentativo di formattare il dischetto prima di visualizzare ERROR-173.

Se il dischetto nuovo ha settori danneggiati, vi raccomandiamo di riportarlo al fornitore per farvelo sostituire. Può darsi che i dischetti che non siano ATARI non abbiano la qualità adatta a lavorare con le unità a dischi ATARI.

Avvertenza: formattando un dischetto si perdono tutti i file esistenti sul dischetto. Se formattate un dischetto formattato II ATARI 810 (CX8111), perderete la maggior velocità di un dischetto preformattato. Usate invece l'opzione D. DELETE FILE(S).

SELECT ITEM OR RETURN FOR MENU
I **RETURN**
WHICH DRIVE TO FORMAT?
1 **RETURN**
TYPE "Y" TO FORMAT DISK 1
Y **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-8 Uso dell'opzione Format Disk

J. DUPLICATE DISK

Usate questa opzione per creare un duplicato esatto di ogni dischetto. Potete usare questa opzione con una singola unità a dischi scambiando manualmente il dischetto sorgente (il dischetto che contiene i file) con il dischetto destinazione (il dischetto sul quale verranno copiati i file) finché il processo di duplicazione è completo. Potete anche usare questa opzione con un sistema di più unità a dischi mettendo i dischetti sorgente e destinazione in due drive diversi e lasciando procedere automaticamente il processo di duplicazione.

Il processo di duplicazione è una tecnica di copiatura settore per settore. Questo significa non solo che tutti i file vengono copiati dal dischetto sorgente al dischetto destinazione ma anche che essi sono messi negli stessi settori su entrambi i dischetti. Viene anche copiata la lista dei file (disk directory) da un dischetto all'altro; per questa ragione, ogni file precedentemente memorizzato sul dischetto destinazione, alla fine del processo di duplicazione sarà stato distrutto.

Il dischetto sorgente deve essere un dischetto formattato dal DOS II, se tentate di usare un dischetto DOS I, otterrete un messaggio di errore. Il dischetto destinazione può essere un qualsiasi dischetto formattato ATARI. Ciò significa che potete usare sia un dischetto preformattato II ATARI (CX8111) che un dischetto formattato

sulla vostra unità a dischi dal DOS I o dal DOS II. Se usate un dischetto vecchio come destinazione, assicuratevi che nessuno dei file su di esso vi interessi, perché il processo di duplicazione li cancellerà scrivendovi sopra i nuovi file.

Ricordate, non potete duplicare un dischetto formattato dal DOS I né potete duplicare un dischetto fra un'unità a dischi ATARI 810 e una ATARI 815; se tentate di farlo otterrete un messaggio d'errore.

Potete comunque realizzare un processo di duplicazione da una unità doppia a dischi ATARI 815 ad una unità a dischi ATARI 810 (e viceversa) e dal DOS I al DOS II usando l'istruzione COPY FILE con l'opzione *.* discussa in precedenza. I dati saranno uguali su entrambi i dischetti ma vi saranno differenze sul modo in cui sono memorizzati. Giacché non vi è un vero e proprio processo di duplicazione fra un dischetto DOS I e DOS II o fra una unità a dischi ATARI 810 e una unità doppia a dischi ATARI 815, dovreste sempre salvare i programmi in BASIC o in linguaggio Assembly che si trovano in RAM prima di duplicare un dischetto. Per l'istruzione DUPLICATE DISK non c'è un buffer interno come c'era per l'istruzione COPY FILE, e il MEM.SAV sarà invalidato se date al DOS II il permesso di procedere (e di usare l'area dei programmi). L'opzione DUPLICATE DISK usa sempre l'area dei programmi (dove è memorizzato il programma BASIC residente in RAM) come buffer per muovere i file dal dischetto sorgente al dischetto destinazione quando si usa un solo drive.

DUPLICAZIONE USANDO UNA SOLA UNITÀ A DISCHI

In un sistema ad una sola unità a disco, il drive sorgente e il drive destinazione è sempre il drive 1 (vedere figura 4-9).

Proteggete sempre il vostro dischetto sorgente a titolo di precauzione, così se questi venisse per errore inserito al posto del dischetto destinazione, apparirebbe sullo schermo ERROR-144 e il dischetto sorgente rimarrebbe intatto.

Se battete qualsiasi carattere che non sia Y **RETURN** in risposta al messaggio: TYPE "Y" IF OK TO USE PROGRAM AREA (batti "Y" per consentire l'uso dell'area di programma) il programma si ferma e appare il messaggio SELECT ITEM OR RETURN FOR MENU.

La figura 4-9 è un esempio di duplicazione usando una singola unità a dischi.

SELECT ITEM OR RETURN FOR MENU
J **RETURN**
DUP DISK—SOURCE, DEST DRIVES?
1,1 **RETURN**
INSERT SOURCE DISK, TYPE RETURN
TYPE "Y" IF OK TO USE PROGRAM AREA?
CAUTION: A "Y" INVALIDATES MEM.SAV
Y **RETURN**
INSERT DESTINATION DISK, TYPE RETURN
RETURN
SELECT ITEM OR RETURN FOR MENU

Figura 4-9 Uso dell'opzione Duplicate Disk con una sola unità a dischi

Nota: il numero di volte che il programma DUP vi richiede di inserire i dischetti sorgente e destinazione, dipende dal numero e dalle dimensioni del file (o dei file) da duplicare per un dato sistema. Per copiare un dischetto pieno, un sistema a 48 K potrebbe richiedere solo due inserimenti, invece un sistema a 16 K potrebbe richiedere 5 o 6 inserimenti.

DUPLICAZIONE USANDO PIÙ UNITÀ A DISCHI

Se state usando sia l'unità a dischi ATARI 810 che l'ATARI 815, assicuratevi di distinguere fra i file memorizzati usando la singola densità e quelli usando la doppia densità quando etichettate i dischetti. Questo vi eviterà di usarli nell'unità a dischi sbagliata.

Anche con un sistema a più unità a dischi, è necessario salvare i programmi BASIC residenti in RAM poiché l'area dei programmi verrà alterata e il MEM.SAV sarà invalidato. Notate che il dischetto sorgente è inserito nel Drive 1 e il dischetto destinazione nel Drive 2 (figura 4-10). Potete usare qualsiasi coppia dello stesso modello di unità a dischi.

Il cursore rimane sullo schermo durante il processo di duplicazione. Questo processo può durare diversi minuti se il dischetto sorgente è quasi pieno.

SELECT ITEM OR RETURN FOR MENU
J **RETURN**
DUP DISK—SOURCE, DEST DRIVES
1,2 **RETURN**
INSERT BOTH DISKS, TYPE RETURN
RETURN
TYPE "Y" IF OK TO USE PROGRAM AREA
CAUTION: A "Y" INVALIDATES MEM.SAV
Y **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-10 Uso dell'opzione Duplicate Disk con un'unità doppia o più unità a dischi

K. BINARY SAVE

Nota: questa opzione non sarà probabilmente usata da un utente principiante del Personal Computer ATARI. Se non capite i numeri esadecimali e non avete una certa conoscenza del linguaggio Assembly, potete fare a meno di leggere le informazioni oltre il primo esempio.

Usate questa selezione dal Menu per salvare i contenuti delle locazioni di memoria nel file oggetto in formato binario. Anche i programmi scritti usando il Cartridge Assembler Editor hanno questo formato. I parametri per questa opzione START, END, INIT, RUN sono numeri esadecimali. Gli indirizzi START (inizio) e END (fine) sono parametri richiesti per ogni file o programma binario. Gli indirizzi INIT (inizializza) e RUN (esegui) sono perametri opzionali che permettono di far eseguire ogni programma al caricamento. Vedere esempi 2, 3 e 4.

Nell'esempio qui sotto, viene salvato un file su un dischetto nel Drive 1, questo file sarà chiamato BINFIL.OBJ con l'indirizzo di partenza 3C00 e l'indirizzo di fine 5BFF. La figura 4-11 è un esempio dell'uso del BINARY SAVE.

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
K **RETURN**
SAVE—GIVE FILE, START, END, INIT, RUN
BINFIL.OBJ, 3C00, 5BFF **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-11 L'uso più elementare di Binary Save

INFORMAZIONI PIÙ DETTAGLIATE RIGUARDO AI PARAMETRI OPZIONALI

Tutti i file binari, come quelli che create con l'opzione BINARY SAVE o con il cartridge Assembler Editor, hanno un'intestazione a 6 byte che precede il file (vedere figura 4-12). Dai dati mostrati nella tabella, potete facilmente ricavare l'indirizzo di inizio e l'indirizzo finale che sono stati usati in figura 4-12.

Byte #	Numero decimale	Numero esadecimale	Descrizione
#1	255	FF	Codice d'identificazione del
#2	255	FF	file binario
#3	0	00	Indirizzo d'inizio (LSB)
#4	60	3C	(MSB)
#5	255	FF	Indirizzo di fine (LSB)
#6	91	5B	(MSB)
			Segmento del file di dati contente 8191 (dec.) bytes di dati.

Figura 4-12 Tabella dell'header a 6 byte per il Binary Save

I due parametri opzionali, INIT e RUN, offrono l'opportunità di fare eseguire automaticamente il file contenente un programma in linguaggio Assembly dopo il caricamento. Un file che fa uso di uno o entrambi questi parametri è chiamato file "carica e parti" (load-and-go). Un file che non contiene parametri per questi dati, è chiamato file "carica" (load), giacché verrà caricato nel computer ma non sarà eseguito finché non verrà data l'istruzione M. RUN AT ADDRESS.

In generale, il parametro RUN definisce il punto in un programma nel quale inizierà l'esecuzione non appena l'intero file sia stato caricato nella RAM (cioè quando viene raggiunta la fine del file). Per questa ragione ci può essere solo un indirizzo di partenza (RUN Address) anche se un file è composto. Per esempio, un file può essere creato unendo diversi piccoli file ognuno dei quali avente il suo indirizzo di partenza. In questo caso, solo l'ultimo indirizzo RUN sarebbe eseguito**.

Se è specificato un indirizzo INIT, non appena il presente indirizzo viene caricato nella RAM, il codice a cui punta sarà eseguito. Questo è vero anche se il file è fatto di diversi file "carica e parti" uniti insieme. In tal caso, ogni segmento "carica e parti" che abbia uno specifico indirizzo INIT sarà eseguito quando viene caricato

l'indirizzo INIT. Dunque, ogni segmento verrebbe caricato ed eseguito prima che venga caricato il segmento successivo ecc.* L'esecuzione del codice puntato da un indirizzo INIT precede sempre l'esecuzione di ogni codice puntato da un indirizzo RUN.

I file creati dal cartridge Assembler Editor che usano l'opzione "carica e parti" possono essere memorizzati negli indirizzi INIT e RUN desiderati, nel vostro codice, seguiti dal codice da controllare. L'indirizzo RUN è sempre alle locazioni 2E0 (parte bassa) e 2E1 (parte alta) esadecimali. L'indirizzo INIT è sempre memorizzato alla locazione 2E2 (parte bassa) e 2E3 (parte alta) esadecimali. Ricordate: l'indirizzo INIT è eseguito non appena viene caricato, quindi il codice cui punta deve essere già stato caricato.

Nota: IOCB #1 è aperto durante l'esecuzione del codice puntato dall'indirizzo INIT. Per questa ragione non è disponibile e non deve essere alterato dall'esecuzione del programma.

**Un RTS (RETURN) alla fine di un programma farà sempre ritornare il controllo al DOS II.

*Ogni segmento di codice deve terminare con un RTS (RETURN) se il segmento successivo deve essere caricato o se il controllo deve ritornare al DOS II.

USO DEL BINARY SAVE CON PARAMETRI OPZIONALI

L'esempio in figura 4-13 illustra un programma in linguaggio Assembly che usa un'area di dati che deve essere inizializzata prima che il programma principale possa usarla. Si supponga che il codice di inizializzazione risieda dall'indirizzo 4000 (hex) al 41FF (hex) e il programma principale dal 4200 (hex) al 4FFF (hex). Per gli scopi di questo esempio si assuma che sia il codice di inizializzazione che il programma principale contengano codici eseguibili e che il codice d'inizializzazione termini con un RTS (RETURN).

Nell'esempio che segue, assumiamo che il programma LAGPRG.OBJ sia già in memoria.

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
K **RETURN**
SAVE—GIVE FILE, START, END, INIT, RUN
LAGPRG.OBJ, 4000, 4FFF, 4000, 4200 **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-13 Uso del Binary Save con parametri opzionali

Caricando questo file in memoria avverrà questo:

1. La memoria da 4000 a 4FFF viene riempita dal programma.
2. L'indirizzo INIT 4000 (hex) è caricato nelle locazioni di memoria 2E2 e 2E3 (hex).
3. Il programma di inizializzazione da 4000 a 41FF viene eseguito.

4. L'indirizzo RUN 4200 (hex) è memorizzato alle locazioni 2E0 e 2E1 (hex).

5. Il programma principale da 4200 a 4FFF comincia l'esecuzione e la continua finché viene eseguito un RETURN (RTS) o viene premuto **SISTEM RESET** o **BREAK**.

Nel caso di un file composto, il risultato è più complicato in relazione a come sono stati creati i nuovi file aggiunti. Il prossimo paragrafo illustra diversi casi in cui sono stati fusi più file.

STRUTTURA DI UN FILE COMPOSTO

Prima di considerare il prossimo esempio, esaminiamo la struttura di un file composto (compound file). Un file composto è costituito da vari file binari che sono stati uniti. Ciò può essere fatto in due modi; un modo è usare l'istruzione C. COPY FILE con la sua opzione d'unione. Un file composto creato in questo modo non è compatibile con il caricatore Assembler Editor, tuttavia può essere caricato con l'opzione L. BINARY LOAD (caricamento binario) del DOS II. Se deve essere compatibile con il cartridge si deve usare l'opzione K. BINARY SAVE. I due tipi di file sono illustrati in appendice I. La sola vera differenza è che, quando viene creato un file composto usando l'opzione C. COPY FILE, il codice d'identificazione FFFF (hex) è incluso con ogni segmento.

Quando è usata l'opzione K. BINARY SAVE, i codici d'identificazione addizionali, per ogni segmento (dopo il primo) NON sono inclusi nel file finale. Questa è la sola forma di file composta compatibile con il cartridge Assembler Editor. L'opzione L. BINARY LOAD del DOS II, tuttavia, è compatibile con entrambi i tipi di file composti.

Ora consideriamo cosa succede quando un file composto come questo viene caricato, supponendo che siano stati specificati vari indirizzi INIT e RUN prima della fusione. (Vi aiuterà pensare gli indirizzi INIT e RUN come facenti parte dei dati in ogni segmento, cosa che in realtà sono).

Esempio 3:

Supponete di avere tre file, ognuno dei quali abbia un indirizzo RUN e nessun indirizzo INIT incluso nei suoi dati. L'esempio in figura 4-14 mostra un modo in cui può essere creato un file come questo.

SELECT ITEM OR RETURN FOR MENU
K **RETURN**
SAVE FILE—GIVE FILE, START, END, INIT, RUN
PART1.OBJ, 2000, 21FF,, 2000 **RETURN**
SELECT ITEM OR RETURN FOR MENU

SELECT ITEM
K **RETURN**
SAVE ITEM—OR RETURN—FOR MENU
PART2.OBJ/A, 2200, 23FF,, 2200 **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-14 Uso del Binary Save per salvare un file composto

Gli altri due file, PART2.OBJ e PART3.OBJ che sono stati creati allo stesso modo di PART1.OBJ, possono essere uniti in WHOLE.OBJ usando le opzioni K. BINARY SAVE o C. COPY FILE con l'opzione di unione. Cosa succede ora, quando questo nuovo file viene caricato?

1. PART1.OBJ viene caricato ma non eseguito (niente INIT).
2. L'indirizzo RUN per PART1.OBJ viene memorizzato in 2E0 e 2E1.
3. PART2.OBJ viene caricato ma non eseguito (niente INIT).
4. L'indirizzo RUN per PART2.OBJ viene memorizzato in 2E0 e 2E1 cancellando l'indirizzo RUN di PART1.OBJ.
5. PART3.OBJ viene caricato ma non eseguito (niente INIT).
6. L'indirizzo RUN per PART3.OBJ è memorizzato in 2E0 e 2E1 cancellando l'indirizzo RUN di PART2.OBJ.
7. L'esecuzione comincia all'indirizzo RUN di PART3.OBJ visto che ora siamo alla fine del file.

Esempio 4:

Per un altro esempio di file composto, considerate un file di tre segmenti: BIGFILE.OBJ. Supponete che ogni segmento sia caricato in una differente area di memoria e che

SEG1.OBJ abbia un indirizzo INIT ma non un indirizzo RUN
SEG2.OBJ non abbia indirizzi INIT o RUN
SEG3.OBJ abbia un indirizzo INIT e un indirizzo RUN per SEG2.OBJ e, in aggiunta, sia caricata sopra SEG1.OBJ.

Quando viene caricato BIGFILE.OBJ avvengono le seguenti cose:

SEG1.OBJ viene caricato

SEG1.OBJ viene eseguito partendo dal suo indirizzo INIT

SEG2.OBJ viene caricato

SEG3.OBJ viene caricato sopra a SEG1.OBJ

SEG3.OBJ viene eseguito, partendo dal suo indirizzo INIT

SEG2.OBJ viene eseguito partendo dall'indirizzo specificato in SEG3.OBJ

È chiaro che questa opzione vi dà la possibilità di creare grandi file che vengono caricati ed eseguiti immediatamente.

Esempio 5:

Per convertire un file esistente che si carica solamente in un file che si carica e parte, potete caricare il file in memoria e quindi salvarlo sotto altro nome usando l'opzione K. BINARY SAVE. Questo pone qualche problema: potreste a volte dimenticare l'indirizzo finale che il file occupa, o il file potrebbe essere composto da segmenti non necessariamente consecutivi in memoria. Potete evitare questi problemi usando la procedura mostrata nell'esempio successivo. Questo esempio illustra un file a solo caricamento con un indirizzo di partenza di 4000 (hex) che è trasformato in un file "carica e parti".

Nella figura 4-15 un file di un byte, localizzato all'indirizzo FF00 (nella ROM O.S.) è aggiunto alla fine del file LOADFIL.OBJ. Siccome l'indirizzo di inizio di questo file è lo stesso indirizzo al quale normalmente il file a solo caricamento parte, questi inizia l'esecuzione non appena l'intero file è caricato in RAM.

SELECT ITEM OR RETURN FOR MENU
K **RETURN**
SAVE FILE—GIVE FILE, START, END, INIT, RUN
LOADFIL.OBJ/A, FF00, FF00,, 4000
SELECT ITEM OR RETURN FOR MENU

Figura 4-15 Conversione di un file esistente a solo caricamento in un file a caricamento e partenza

L. BINARY LOAD

Nota: questa istruzione non sarà probabilmente usata da un utente principiante dei Personal Computer ATARI.

Usate questa opzione per caricare nella RAM un file binario in linguaggio Assembly che è stato salvato in precedenza con l'opzione K. del Menu o creato dal cartridge Assembler Editor. Se l'indirizzo RUN o l'indirizzo INIT erano stati uniti al file nelle locazioni 2E0 e 2E1 o 2E2 e 2E3 il file partirà automaticamente dopo essere stato caricato. In un file "carica e parti" gli indirizzi RUN e INIT sono ignorati quando battete /N dopo il nome del file (filename) (vedere esempio 1 in figura 4-16). Il file può essere fatto partire usando l'opzione del Menu RUN AT ADDRESS (esegui all'indirizzo).

L'uso di questa istruzione senza l'opzione /N è mostrato nel secondo esempio in figura 4-16. Siccome questo file aveva l'indirizzo d'inizio nelle locazioni 2E0 e 2E1 (vedere esempio 1 per il K. BINARY SAVE), il file comincerà l'esecuzione non appena caricato.

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
L **RETURN**
LOAD FROM WHAT FILE?
MYFILE.OBJ/N **RETURN**
SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
L **RETURN**
LOAD FROM WHAT FILE?
BINFIL.OBJ **RETURN**

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
L **RETURN**
LOAD FROM WHAT FILE?
MACHL.OBJ **RETURN**
SELECT ITEM OR RETURN FOR MENU

Figura 4-16 Uso dell'opzione Binary Load

L'esempio 3 in figura 4-16 illustra un file chiamato MACHL.OBJ che non ha un indirizzo RUN o INIT. In questo caso, il messaggio SELECT ITEM OR RETURN FOR MENU apparirà non appena il file termina il caricamento.

Per eseguire un file che non ha indirizzi RUN e INIT vedere l'opzione del Menu: M. RUN AT ADDRESS (parti all'indirizzo).

M. RUN AT ADDRESS

Nota: questa opzione non verrà probabilmente usata da un utente principiante dei Personal Computer ATARI.

Usate questa opzione per inserire l'indirizzo esadecimale di un programma oggetto caricato in RAM con l'opzione BINARY LOAD. Questa opzione è usata quando l'indirizzo di inizio non era stato specificato nel file oggetto.

Nella figura 4-17 le istruzioni della locazione esadecimale 3000 saranno eseguite. Siate molto cauti quando inserite questi indirizzi esadecimali. Se inserite un indirizzo che non contiene un codice eseguibile, si avranno problemi. Per esempio potreste bloccare il sistema rendendo necessario spegnerlo e riaccenderlo.

SELECT ITEM OR RETURN FOR MENU
M **RETURN**
RUN FROM WHAT ADDRESS?
3000 **RETURN**

Figura 4-17 Uso dell'opzione Run at Address

N. CREATE MEM.SAV

Questa opzione vi permette di creare un file su un dischetto, chiamato MEM.SAV, nel quale il contenuto della memoria riservata all'utente viene salvata quando si chiama il DOS. Quando battete **DOS RETURN**, il computer salva il programma residente in RAM (se c'è) nel file MEM.SAV, prima di caricare in memoria il file DUP.SYS. Quando terminate di usare le opzioni DOS, semplicemente ripassate il controllo al cartridge battendo B **RETURN** e il MEM.SAV automaticamente ricaricherà il vostro programma in RAM. Se non avete un cartridge inserito, battere **B** non ha alcun effetto; vedrete apparire il messaggio: SELECT ITEM OR RETURN FOR MENU.

Dovete aver cura di non permettere al DOS di usare tutta la memoria riservata all'utente, quando volete le opzioni COPY FILE, DUPLICATE FILE o DUPLICATE DISK per salvare i dati esistenti. DOS non sa se TUTTO o solo parte del vostro programma è stato salvato in MEM.SAV. Quando DOS utilizza tutta la memoria disponibile all'utente, viene automaticamente invalidato il file MEM.SAV. Se questo succede, il vostro programma non sarà ricaricato quando il controllo verrà passato al cartridge.

SELECT ITEM OR RETURN FOR MENU
N **RETURN**
TYPE "Y" TO CREATE MEM.SAV

Y **RETURN**
SELECT ITEM OR RETURN FOR MENU

Fig. 4-18 Creare un file MEM.SAV

Se tentate di usare questa opzione per creare un file MEM.SAV su un dischetto sul quale è già presente, lo schermo visualizzerà il messaggio: MEM.SAV file already exist (il file MEM.SAV esiste già) seguito dal messaggio SELECT ITEM OR RETURN FOR MENU. La figura 4-18 illustra i passi per creare un file MEM.SAV su un dischetto inserito nel drive 1. Notate che i file MEM.SAV possono essere creati solo su un dischetto nel drive 1.

PERCHÈ AVERE UN FILE MEM.SAV?

Questo speciale file vi permette di salvare temporaneamente il programma in RAM in un file speciale sul dischetto. Il MEM.SAV (che richiede 45 settori) deve essere sul dischetto inserito nel drive 1. Questo dischetto non deve essere protetto se MEM.SAV deve lavorare. Quando MEM.SAV esiste sul dischetto, l'area di memoria che dovrà essere cancellata all'ingresso del DUP.SYS, verrà memorizzata in MEM.SAV ogni volta che viene chiamato il DOS. In sostanza viene effettuata un'operazione di scambio di contenuti, tramite un'"espansione" dell'area dei programmi. Questo scambio impiega circa 21 secondi. Quando il controllo del sistema ritorna al cartridge, il DUP.SYS viene cancellato dai contenuti di MEM.SAV che vengono ricaricati automaticamente in RAM. Questa operazione richiede circa 7 secondi.

Se state lavorando su un programma BASIC e avete bisogno, per qualsiasi ragione, di tornare al DOS, potete farlo usando MEM.SAV senza dover salvare il vostro programma su dischetto e ricaricarlo. Quando terminate di usare il DOS e restituite il controllo del sistema al cartridge, il file MEM.SAV viene automaticamente ricaricato in memoria e il vostro programma BASIC è rimemorizzato nella RAM.

Un esempio di uso di MEM.SAV è illustrato nella figura 4-19.

1. Battete **LOAD "D:MYPROG.BAS" RETURN**.
2. Listate il vostro programma e battete **RUN RETURN**.
3. Esso funziona e volete cambiare nome al file originale per tenerlo come copia.
4. Battete **DOS RETURN**.
5. Fate la vostra selezione (**E** per RENAME FILE) e cambiate il nome del vostro file in: MYPROG.OLD.
6. Battete **B RETURN** per tornare al BASIC. Con l'aiuto di MEM.SAV la versione modificata di MYPROG.BAS è caricata automaticamente in RAM.
7. Battete SAVE "D:MYPROG.BAS" **RETURN** per salvare il vostro programma modificato, sotto il nome originale.

Fig. 4-19 Esempio dell'uso di MEM.SAV

USO DEL MEM.SAV PER SCRIVERE PROGRAMMI IN LINGUAGGIO ASSEMBLY

Il file MEM.SAV vi permette anche di scrivere programmi in linguaggio Assembly (o caricare dati binari) che condividono l'area di programmi con DUP.SYS. Questo significa che siete liberi di scrivere programmi o caricare dati nell'area da LOMEM (limite inferiore della memoria disponibile, varia in relazione al numero di drive nel sistema e al numero di file che possono essere aperti simultaneamente) a HIMEM (limite superiore della memoria disponibile, che varia in relazione al modo grafico in cui vi trovate). Vedere appendice D, mappa di memoria.

Esempio:

Supponete di avere un file binario che volete sia eseguito automaticamente non appena viene caricato. Questo genere di file è detto "carica e parti"; l'indirizzo di partenza è già programmato e non avrete bisogno di selezionare l'opzione RUN AT ADDRESS. In questo caso, non è necessario avere un file MEM.SAV nel dischetto. Siccome il file è "carica e parti" sarà semplicemente caricato e quindi comincerà l'esecuzione. Il modo più sicuro di tornare al DOS è riaccendere il computer. Se non avete invaso l'area del DUP.SYS durante l'esecuzione del vostro file binario, potete ritornare al DOS semplicemente eseguendo l'istruzione RETURN (RTS) nel vostro programma. Se, invece, il programma ha invaso l'area del DUP.SYS cancellandolo, il DOS ne prenderà atto e ricaricherà ed eseguirà automaticamente DUP.SYS quando il RETURN nel vostro programma verrà eseguito.

Avvertenza: se l'esecuzione del vostro file "carica e parti" scrive in un'area sotto LOMEM usata dal DOS.SYS, DUP.SYS o in un'area RAM usata dal sistema operativo, il RETURN (RTS) del vostro programma può lasciare il computer in uno stato indefinito. Se ciò capita, dovete riaccendere il computer per riprendere.

USO DEL MEM.SAV PER CARICARE FILE BINARI

Questo paragrafo tratta il caricamento di un file binario che non viene eseguito al caricamento o di un file che contiene dati per un altro programma. Se il caricamento del vostro programma non si sovrappone a nessuna parte dell'area DUP. SYS, allora il file MEM.SAV non è richiesto.

Se il file che viene caricato si sovrappone a parte del file DUP.SYS, dovrete avere un file MEM.SAV sul dischetto nel Drive 1 perché il caricamento abbia successo. Dopo l'esecuzione dell'opzione L. LOAD BINARY SAVE avviene:

1. Usate la selezione LOAD BINARY FILE per caricare il vostro file.
2. Il vostro originale MEM.SAV è caricato dal disco nella memoria sovrapponendosi e invalidando il DUP.SYS.
3. Il vostro file è caricato sopra l'originale MEM.SAV modificando parte o tutto il file MEM.SAV originale.
4. Il vostro nuovo file MEM.SAV nella RAM è salvato nell'area MEM.SAV sul dischetto.
5. DUP.SYS è ricaricato dal disco nella memoria.
6. Rimanete nel modo DOS finché non scegliete:

RUN CARTRIDGE	il vostro file è caricato in memoria dal MEM.SAV e il controllo ritorna al cartridge BASIC o Assembly.
RUN AT ADDRESS	il vostro file è caricato nella memoria dal MEM.SAV e iniziate l'esecuzione dell'indirizzo che voi specificate.
LOAD BINARY FILE	quando volete caricare un file "carica e parti". In questo caso, se anche il nuovo file si sovrappone a parte del DUP.SYS ma non al file originale, allora sia MEM.SAV che il vostro nuovo file saranno in memoria quando il caricamento è completo. Se il nuovo file non si sovrappone affatto al DUP.SYS allora il caricamento sarà completo con solo il nuovo file caricato in RAM. Siccome il nuovo file è un "carica e parti" e viene caricato sia che si sovrapponga a DUP.SYS che in caso contrario, rimarrete sotto il controllo di questo file finché viene eseguito un RETURN (RTS).

Nota: se volete avere due file in memoria simultaneamente, uno dei quali risieda interamente o in parte nell'area DUP.SYS e l'altro completamente fuori dall'area DUP.SYS, il modo più semplice per ottenere ciò è unire i due file in un file solo e quindi caricare il file appena creato.

O. DUPLICATE FILE

Questa opzione è usata se avete solo una unità a dischi e volete copiare un file da un dischetto ad un altro. Ricordate che una singola unità a dischi deve sempre avere il numero 1. Siccome c'è una sola unità a dischi, dovete inserire e togliere manualmente i dischetti sorgente e destinazione. Se un file è molto lungo potreste dover alternare i dischetti sorgente e destinazione diverse volte prima che l'operazione di duplicazione sia completa. Permettendo al DOS di usare la memoria riservata all'utente, si ridurranno il numero di sostituzioni richieste per copiare un lungo file, tuttavia dovete ricordarvi che ciò implica che il MEM.SAV sarà invalidato.

I caratteri jolly (wild cards) sono ammessi in questa opzione. Nell'esempio 2 noterete che anche se date il permesso al DOS di utilizzare la memoria riservata all'utente (con un filename costituito da un carattere jolly), i vostri file sono copiati uno alla volta. Dovrete alternare i dischetti almeno una volta per ogni file che volete copiare.

Il secondo esempio illustra l'uso dei caratteri jolly per copiare i file aventi cinque lettere e inizianti con TEST da un dischetto a un'altro. In questo esempio si suppone che il dischetto sorgente abbia solo due file con il nome che soddisfa TEST?

Nell'esempio 3 sia i nomi che le estensioni sono stati sostituiti con caratteri jolly. Il DOS copierà quindi tutti i file eccetto quelli che hanno l'estensione .SYS. In questo esempio si assume che ci siano solo tre file da copiare: MEM.SAV, TEST1 e TEST2.

Esempio 1:

SELECT ITEM OR RETURN FOR MENU
O RETURN

NAME OF FILE TO MOVE?
DOSEX. BAS **RETURN**
TYPE "Y" IF OK TO USE PROGRAM AREA
CAUTION: A "Y" INVALIDATES MEM.SAV
Y **RETURN**
INSERT SOURCE DISK, TYPE RETURN
RETURN
INSERT DESTINATION DISK, TYPE RETURN
RETURN
SELECT ITEM OR RETURN FOR MENU

Esempio 2:

SELECT ITEM OR RETURN FOR MENU
O **RETURN**
NAME OF FILE TO MOVE?
TEST? **RETURN**
TYPE "Y" IF OK TO USE PROGRAM AREA
CAUTION: A "Y" INVALIDATES MEM.SAV
Y **RETURN**
INSERT SOURCE DISK, TYPE RETURN
RETURN

COPYING---D1: TEST1
INSERT DESTINATION DISK, TYPE RETURN
RETURN

INSERT SOURCE DISK, TYPE RETURN
RETURN

COPYING---D1: TEST2
INSERT DESTINATION DISK, TYPE RETURN
RETURN
INSERT SOURCE DISK, TYPE RETURN
RETURN
SELECT ITEM OR RETURN FOR MENU

Esempio 3:

SELECT ITEM OR RETURN FOR MENU
O **RETURN**
NAME OF FILE TO MOVE?
* * **RETURN**
TYPE "Y" IF OK TO USE PROGRAM AREA
CAUTION: A "Y" INVALIDATES MEM.SAV
Y **RETURN**
INSERT SOURCE DISK, TYPE RETURN
RETURN

COPYING---D1: MEM.SAV
INSERT DESTINATION DISK, TYPE RETURN
RETURN
INSERT SOURCE DISK, TYPE RETURN
RETURN
COPYING---D1: TEST1
INSERT DESTINATION DISK, TYPE RETURN
RETURN

```
INSERT SOURCE DISK, TYPE RETURN
RETURN
  COPYING---D1: TEST2
INSERT DESTINATION DISK, TYPE RETURN
RETURN
INSERT SOURCE DISK, TYPE RETURN
RETURN
SELECT ITEM OR RETURN FOR MENU
```

Figura 4-20 Uso dell'opzione Duplicate File

5

ULTERIORI INFORMAZIONI

ISTRUZIONI BASIC USATE COL DOS

Prima di descrivere le istruzioni BASIC usate col DOS II, è necessario che sappiate come le istruzioni agiscono sui programmi memorizzati e richiamati. I paragrafi seguenti spiegano i due tipi di file che possono contenere programmi BASIC.

FILE "TOKENIZZATI" E NON "TOKENIZZATI"

I file del primo tipo, detto non "tokenizzato", contengono caratteri di testo in standard ATASCII così da apparire come un listato di programma BASIC. Questi programmi non conservano la loro tabella di simboli (symbol table) ogni volta che sono caricati e salvati. La tabella dei simboli associa il nome della variabile con la locazione di memoria nella quale è contenuto il valore di quella variabile. Per memorizzare e richiamare un file nella sua forma non "tokenizzata" usate le istruzioni LIST e ENTER.

Il secondo tipo, detto "tokenizzato" è una versione condensata di un programma BASIC. Esso usa "tokens" di un byte invece dei caratteri ATASCII per rappresentare le istruzioni BASIC.

I programmi tokenizzati sono trasferiti fra l'unità a dischi e la console del Computer dalle istruzioni SAVE e LOAD. Le versioni tokenizzate di un file sono generalmente più corte di quelle non tokenizzate; per questa ragione, molti programmatori, preferiscono memorizzare i loro programmi finali nella forma tokenizzata perché si caricano più velocemente e occupano minor spazio nel dischetto. Una versione tokenizzata conserva la sua tabella dei simboli ogni volta che viene ripreso.

LOAD (LO.)

Formato: LOAD specifica del file
Esempio: LOAD "D1: DOSEX.BAS"
RETURN

Questa istruzione è usata per caricare un file da un particolare dischetto in una unità a disco nell'area dei programmi in RAM. Per usare questa istruzione per caricare un file chiamato DOXES.BAS, il file (DOXES.BAS) deve essere stato salvato in precedenza usando l'istruzione BASIC. SAVE. Questa istruzione carica solo la versione tokenizzata di un programma.

Questa istruzione può essere anche usata per "concatenare" i programmi (figura 5-1). Se avete un programma che è troppo grosso per essere eseguito nella vostra RAM disponibile potete usare l'istruzione LOAD come ultima linea del primo programma (figura 5-1). Quindi, quando il programma incontra l'istruzione LOAD leggerà automaticamente la parte successiva del programma dal dischetto. Tuttavia

il secondo programma deve essere in grado di lavorare da solo, senza dipendere da nessuna variabile o dato nella RAM dal primo programma. Il programma caricato non sarà eseguito finché non battete RUN **RETURN** nel qual caso il programma precedente e ogni variabile saranno cancellate. (vedere RUN per un altro esempio).

```
100 REM    Programma di concatenamento
110 LOAD   "D1: CHAIN.BAS"
```

Figura 5-1   Esempio di concatenamento di programma

SAVE (S.)

Formato: SAVE specifica del file
Esempio: SAVE "D1/EXAMP2.BAS"
RETURN

Questa istruzione fa salvare un programma sul dischetto con il nome specificato nell'istruzione. SAVE è l'istruzione complementare di LOAD e memorizza i programmi in forma tokenizzata.

LIST (L.)

Formati: LIST specifica, n.linea, n. periferico
Esempi: LIST "D:DATFIL.LST"
LIST "P:"
LIST "P:", 10, 100

L'uso di questa istruzione in BASIC è molto simile all'istruzione SAVE visto che può prendere un programma dalla RAM e salvarlo in un particolare drive con qualsivoglia nome (come si vede dal primo esempio). Tuttavia, il programma è memorizzato in testo ATASCII standard e non in Tokens. Differenze nel formato dei dati immagazzinati, permettono a LIST di essere molto più flessibili di SAVE. Come è mostrato negli esempi sottostanti, è possibile specificare una singola periferica (esempio P:, E:, C:, D:, D2; ecc.) o i numeri di linea da listare per una determinata periferica (esempio "P:", 100, 200).

Se non specificate alcuna periferica dopo l'istruzione LIST, tutti i numeri di linea inseriti verranno mostrati sullo schermo. Quando non si specifica la periferica, in questa istruzione, viene sottinteso lo schermo (E:).

Riassumendo, la differenza principale fra LIST e SAVE è che LIST sposta il testo in standard ATASCII in diverse periferiche, mentre SAVE può solo salvare programmi BASIC Tokenizzati su un dischetto.

ENTER (E.)

Formato: ENTER specifica di un file
Esempio: ENTER "D. LIST2.LST"

Questa istruzione sposta il file col nome specificato dal dischetto specificato, nel RAM. Il programma è caricato nella forma non Tokenizzata e viene interpretato all'ingresso dei dati. ENTER, a differenza di LOAD, non distruggerà il programma BASIC-RAM residente ma fonderà (merge) il programma residente in RAM e il file che è stato caricato da disco. Se ci sono linee con lo stesso numero nei due programmi, le linee nel programma entrante sostituiranno quelle del programma RAM-residente.

RUN

Formato: RUN specifica del file
Esempio: RUN "D2: MYFILE.BAS"

Questa istruzione causa il caricamento e l'esecuzione del file scelto. È una combinazione delle istruzioni LOAD e RUN, tuttavia l'istruzione RUN può essere usata solo con file tokenizzati. Dunque, non potete eseguire l'istruzione RUN "D2: LIST.LST".

Per unire programmi e far sì che un secondo segmento venga caricato ed eseguito automaticamente, potete usare l'istruzione RUN "D: filespec" come ultima linea del primo segmento, tuttavia, il secondo programma deve poter funzionare da solo senza dipendere da variabili o dati nella RAM dal primo programma. Prima di lanciare (RUN) il primo segmento, assicuratevi di averlo salvato su un dischetto, poiché l'istruzione RUN farà svanire il primo segmento quando il secondo verrà caricato.

BLOCCHI DI CONTROLLO INPUT/OUTPUT

Un'operazione I/O è controllata da un blocco di controllo I/O (I/O Control Block: IOCB). Un IOCB è una specificazione dell'operazione I/O che consiste nel tipo di I/O, nella lunghezza del buffer e nell'indirizzo del buffer e in altre due variabili di controllo ausiliarie delle quali la seconda è generalmente 0. Il BASIC ATARI predispone otto IOCB e ne dedica tre come segue:

- IOCB #0 è usato dal BASIC per l'I/O di E:
- IOCB #6 è usato dal BASIC per l'I/O di S:
- IOCB #7 è usato dal BASIC per le istruzioni LPRINT, CLOAD e SAVE.

Gli IOCB da #1 a #5 possono essere usati liberamente ma gli IOCB dedicati dovrebbero essere evitati a meno che un programma non faccia uso di uno degli IOCB dedicati. IOCB #0 non può mai essere aperto o chiuso da un programma BASIC.

IOCB CON LE ISTRUZIONI INPUT/OUTPUT

Ogni istruzione input/output deve essere associata ad un IOCB.. Le istruzioni I/O che possono essere usate col DOS II sono le seguenti:

OPEN/CLOSE
INPUT/PRINT
PUT/GET
STATUS
XIO

USO DELLE ISTRUZIONI OPEN/CLOSE

OPEN (O.)

Formato: OPEN #iocb, aexp1, aexp2, specifica di un file
Esempio: 100 OPEN #2, 8, 0, "D1: ATARI800.BAS"

L'istruzione OPEN associa uno specifico IOCB alla periferica appropriata, inizializza ogni variabile relativa al CIO (vedere glossario) e passa alla periferica ogni opzione specifica ad essa. I parametri in questa istruzione sono illustrati in figura 5-2.

#	Carattere obbligatorio inserito dall'utente.
iocb	Un numero fra 1 e 7 che si riferisce ad una periferica o ad un file.
aexp1	Numero che determina il tipo di operazione da compiersi. Codice 4 = operazione di input; posiziona il puntatore del file all'inizio di questo. 6 = operazione di input della lista dei file (disk directory). 8 = operazione di output; posiziona il puntatore del file all'inizio di questo 9 = operazione "end-of-file append" (unione alla fine del file); posiziona il puntatore del file alla fine di questo. Il codice 9 permette un input da programma dall'editor dello schermo senza che l'utente prema **RETURN** 12 = operazione input-output; posiziona il puntatore del file all'inizio di questo.
aexp2	Codice ausiliario dipendente dalla periferica. Un 83 (ASCII S) in questa posizione fa stampare lateralmente l'ATARI 820 Printer; altrimenti è sempre 0 (zero).
file-spec	Identificatore di un file (vedere il capitolo 1).

Figura 5-2 Parametri dell'istruzione OPEN

Nell'esempio, OPEN #2, 8, 0, "D1: ATARI800.BAS", IOCB #2 è aperto per l'output di un file nel Drive 1 chiamato ATARI800.BAS. Se non ci sono file con questo nome nel Drive 1 il DOS lo crea. Se un file con questo nome esiste già, allora l'istruzione OPEN lo distrugge e ne crea un altro. Se l'IOCB è già stato aperto, viene visualizzato ERROR-129 (file già aperto).

CLOSE (CL.)

Formato: CLOSE #iocb
Esempio: 300 CLOSE #2

L'istruzione CLOSE rilascia l'IOCB che è stato aperto in precedenza per le operazioni di lettura/scrittura. Il numero che segue il # deve essere lo stesso numero relativo all'IOCB che è stato usato nell'istruzione OPEN (vedere esempio più sotto). Se l'IOCB è stato già aperto per una periferica e si tenta di aprirlo per un'altra periferica senza chiuderla, viene visualizzato ERROR-129. Lo stesso IOCB non può essere usato per più di una periferica contemporaneamente. Non si ha messaggio d'errore se viene chiuso un file già chiuso.

10 OPEN #1, 8, 0, "D:FIL.BAS"
20 CLOSE #1

Figura 5-3 Esempio di apertura e chiusura di un file

Nota: l'istruzione END chiude tutti i file aperti (eccetto IOCB #0).

USO DELLE ISTRUZIONI INPUT/PRINT

INPUT (I.)

Formato: INPUT [#iocb {,|;}] {avar|svar} [,{avar|svar}...]

Esempi: 100 INPUT #2; X, Y
100 INPUT #2; N$

Questa istruzione è usata per chiedere dati (numerici o stringhe) da una specifica periferica. INPUT è il complemento di PRINT, quando viene usato senza un iocb, si assume che il dato provenga da (E:). INPUT usa i records I/O (vedi PRINT).

```
5 REM **CREAZIONE DI UN FILE DI DATI**
7 REM **OPEN CON 8 CREA UN FILE DI DATI**
10 OPEN #1,8,0,"D:WRITE.DAT"
20 DIM WRT$(60)
30 ? "INSERISCI UNA FRASE NON PIU' LUNGA DI   60 CARATTERI"
35 INPUT WRT$
38 REM **SCRIVE I DATI SUL DISCHETTO**
40 PRINT #1,WRT$
45 REM **CHIUDE IL FILE DI DATI**
50 CLOSE #1
55 REM **APRE IL FILE PER LA LETTURA**
60 OPEN #1,4,0,"D:WRITE.DAT"
65 REM **LEGGE IL DATO DAL DISCHETTO**
70 INPUT #1,WRT$
75 REM **SCRIVE IL DATO**
80 PRINT WRT$
85 REM **CHIUDE IL FILE DI DATI**
90 CLOSE #1
```

Figura 5-4 Programma tipo INPUT/PRINT

In figura 5-4, la linea 35 permette all'utente di battere dati sulla tastiera (la periferica sottointesa). Alla linea 70 l'istruzione INPUT legge il contenuto di una stringa dal file aperto.

PRINT (PR. o ?)

Formato: PRINT {#iocb|[exp]} [{,|;} [exp]...]

Esempi: 100 PRINT #2; X, Y
100 PRINT #2; A$
100 ? C$
100 PRINT "X = ",X

Questa istruzione scrive un'espressione (sia stringa che aritmetica) sulla periferica aperta con lo stesso numero di riferimento IOCB.

Se non viene specificato nessun numero IOCB, il sistema scrive l'espressione sullo schermo, che è la periferica sottintesa. Se l'informazione è diretta a una periferica che non è aperta, appare sullo schermo ERROR-133.

PRINT tratta i cosiddetti records I/O. I records sono set di byte separati dal carattere fine-linea (9B hex). Le dimensioni di un record sono arbitrarie. Le dimensioni di un record possono essere determinate dalla lunghezza di una stringa "stampata" su un file in un dischetto o dal formato di una variabile aritmetica. Può anche essere la lunghezza di una stringa di caratteri inserita dalla tastiera e terminata da **RETURN**.

L'istruzione INPUT non può (generalmente) leggere un record che sia più lungo di 110 caratteri. Se scrivete (PRINT) un record su disco che in seguito vorrete riprendere (INPUT), è meglio limitare la lunghezza del record a 110 caratteri o meno.

ACCESSO DIRETTO CON LE ISTRUZIONI NOTE/POINT

NOTE (NO.)

Formato: NOTE #iocb, avar, avar
Esempio: NOTE #2, A, B

I file sono creati in sequenza e vi si accede normalmente dall'inizio alla fine. Se volete accedere a un file in maniera non sequenziale (diretta), potete leggere il file sequenzialmente e fermarvi al record che desiderate; oppure avete bisogno di un metodo speciale di indirizzamento al record desiderato.

```
1 REM ESEMPIO DELL' ISTRUZIONE NOTE
2 REM QUESTO PROGRAMMA LEGGE LINEE DI
3 REM DATI DALLA TASTIERA E LI MEMO
4 REM RIZZA SU DISCO NEL FILE
5 REM D:DATFIL.DAT.I PUNTATORI SONO
6 REM MEMORIZZATI IN D:POINTS.DAT
20 DIM A$(20)
25 OPEN #1,8,0,"D:DATFIL.DAT"
27 OPEN #2,8,0,"D:POINTS.DAT"
30 REM LEGGE LINEE DI DATI DALLA TAST.
40 INPUT A$
41 LPRINT A$
42 REM SE A$=RETURN ALLORA STOP
45 IF LEN(A$)=0 THEN 100
50 NOTE #1,X,Y
55 REM MEMORIZZA LINEE DI DATI
60 PRINT #1,A$
61 REM MEMORIZZA I PUNTATORI ALL' INI
```

```
62 REM ZIO DELLE LINEE DI DATI.
65 PRINT #2;X;",";Y
70 LPRINT "SETTORE # = ";X,"BYTE # =";Y
90 GOTO 40
95 REM INDICA LA FINE DEL FILE
100 PRINT #2;0;",";0
110 END
```

Figura 5-5 Programma tipo per l'istruzione NOTE

Col primo metodo, si perde molto tempo se il file è grande, perciò il DOS II incorpora NOTE e POINT per darvi la possibilità di accedere a un file in modo casuale. Per giungere ad un record senza passare attraverso ogni record che lo precede, dovete far sapere al computer quale record volete. Questo richiede una "nota" del settore del file; perciò dovete usare l'istruzione NOTE prima di ogni operazione di scrittura e salvate il valore di ritorno in una tabella.

Questa istruzione ottiene il valore del puntatore (file pointer) per il file che usa l'IOCB specificato. Il puntatore del file specifica l'esatta posizione in cui il prossimo byte dovrà essere letto o scritto. Questa istruzione memorizza il numero del settore nella prima variabile aritmetica e il numero del byte nella seconda. Il numero dei settori varia da 1 a 719 e il numero del byte da 0 a 124.

La figura seguente è il risultato dell'esecuzione del programma di figura 5-5, e mostra un modo di usare l'istruzione NOTE per memorizzare gli input da tastiera in una specifica locazione di un file.

Lo stampato in figura 5-5 rappresenta un esempio di programma per l'istruzione NOTE. Nell'esempio di esecuzione, si sono usati numeri, ma è possibile battere qualsiasi stringa per A$ fino a 40 caratteri.

L'esempio è stato eseguito su un dischetto che contiene i file DOS.SYS, DUP.SYS e MEM.SAV. I vostri numeri di settore e di byte possono essere differenti. Nell'esempio sono stati inseriti i numeri: 45, 55, 75, 80, 90, 100, 110.

```
45
SECTOR # = 145      BYTE # = 9
55
SECTOR # = 145      BYTE # = 12
75
SECTOR # = 145      BYTE # = 15
80
SECTOR # = 145      BYTE # = 18
90
SECTOR # = 145      BYTE # = 21
100
SECTOR # = 145      BYTE # = 24
110
SECTOR # = 145      BYTE # = 28
```

Figura 5-6 Esempio di Esecuzione del programma NOTE

POINT (P.)

Formato: POINT # iocb, avar, avar
Esempio: 100 POINT #2, A, B

POINT è il complemento di NOTE: questa istruzione assegna al file pointer un valore arbitrario determinato dalle variabili A e B. POINT è usato per la lettura di una locazione specifica in un file (settore e byte) nella RAM. La prima variabile aritmetica specifica il numero del settore, la seconda il numero del byte nel quale il prossimo byte sarà scritto o letto. Come per l'istruzione NOTE il numero del settore va da 1 a 719 e il numero del byte va da 0 a 125. Se puntate ad un indirizzo esterno al file aperto, otterrete un messaggio d'errore. Il listato (figura 5-7) e l'esempio di esecuzione (figura 5-8) contiene un esempio della istruzione POINT per leggere i dati creati dal programma mostrato come esempio della istruzione NOTE.

Quando viene fatto partire, questo programma stampa l'input da tastiera per settori e per byte nell'ordine inverso di come sono stati scritti sul dischetto.

```
1 REM ESEMPIO DELL' ISTRUZIONE POINT
2 REM QUESTO PROGRAMMA PRENDE IL FILE
3 REM CREATO DA NOTE E STAMPA
4 REM LE LINEE NELL' ORDINE INVERSO
10 DIM B(20,1)
20 DIM A$(40)
25 REM APRE IL FILE DI DATI
30 OPEN #1,4,0,"D:DATFIL.DAT"
35 REM APRE IL FILE DEI PUNTATORI
40 OPEN #2,4,0,"D:POINTS.DAT"
45 REM LEGGE I PUNTATORI IN UN ARRAY
50 FOR I=0 TO 20
60 INPUT #2;X,Y
70 B(I,0)=X:B(I,1)=Y
80 IF X=0 AND Y=0 THEN LAST=I:I=20
90 NEXT I
95 REM SCRIVE IL FILE NELL' ORDINE INV
100 FOR I=LAST-1 TO 0 STEP -1
110 X=B(I,0):Y=B(I,1)
120 POINT #1,X,Y
130 LPRINT "SETTORE # = ";X,"BYTE # = ";Y
140 INPUT #1;A$
150 LPRINT A$
160 NEXT I
```

Figura 5-7 Esempio di programma per l'uso di istruzione POINT

Dopo aver battuto gli esempi di programma NOTE e POINT, battete RUN.

La figura 5-8 è un esempio di esecuzione.

```
SECTOR # = 145          BYTE # = 28
110
SECTOR # = 145          BYTE # = 24
100
SECTOR # = 145          BYTE # = 21
90
SECTOR # = 145          BYTE # = 18
80
SECTOR # = 145          BYTE # = 15
75
SECTOR # = 145          BYTE # = 12
55
SECTOR # = 145          BYTE # = 9
45
```

Figura 5-8 Esempio di esecuzione del programma POINT

USO DELLE ISTRUZIONI PUT E GET

PUT (PU.)

Formato: PUT #iocb, aexp
Esempio: 100 PUT #6, ASC ("A")

L'istruzione PUT scrive un singolo byte (valore da 0 a 255) nella periferica specificata dal numero di IOCB. Nella figura 5-9 l'istruzione PUT è usata per scrivere i numeri che battete in un array dimensionato come A(50). Potete inserire fino a 50 numeri ognuno dei quali deve essere minore di 256. Questa istruzione è usata per creare file di dati o per aggiungere dati ad un file esistente.

```
10 GRAPHICS 0:REM DEMO PUT/GET
20 DIM A(50),A$(10)
30 GRAPHICS 0:? "ESEMPIO DI PROGRAMMA DI PUT E GET SU DISCO":?
40 ? "Dovrai LEGGERE o SCRIVERE ?":INPUT A$:?
50 IF A$="LEGGERE" THEN 170
60 IF A$<>"SCRIVERE" THEN PRINT "?":GOTO 40
70 REM ROUTINE DI SCRITTURA
80 OPEN #1,8,0,"D1:ESEMPIO1.DAT"
90 ? "Inserite un numero minore di 256":INPUT X
95 REM ***SCRIVE IL NUMERO NEL FILE***
100 PUT #1,X
110 IF X=0 THEN CLOSE #1:GOTO 130
120 GOTO 90
```

Figura 5-9 Esempio di programma PUT

GET (GE.)

Formato: GET #iocb, avar
Esempio: 100 GET #2, X

Questa istruzione legge un singolo byte dalla periferica specificata dal numero di IOCB, in una singola variabile. La seconda parte dell'esempio di programma (fig. 5-10) illustra l'istruzione GET. Questa vi permette di rileggere qualsiasi byte memorizzato con l'istruzione PUT.

Notate che INPUT/PRINT e GET/PUT sono tipi di istruzioni INPUT/OUTPUT incompatibili fra di loro. PRINT inserisce il carattere fine-linea (end-of-line EOL) fra i records e INPUT usa questi per determinare un record. GET e PUT scrivono semplicemente singoli byte in un file, senza separarli con EOL. Un file creato usando l'istruzione PUT appare come un grande record, a meno che non abbiate introdotto un carattere EOL (9B hex) tra gli altri.

```
130 GRAPHICS 0:? :? "Adesso leggo i dati dal file ?":INPUT A$
140 IF A$="NO" THEN END
150 IF A$<>"SI" THEN 130
160 REM ROUTINE DI LETTURA
170 OPEN #2,4,0,"D1:ESEMPIO1.DAT"
180 FOR E=1 TO 50
185 REM LEGGE I NUMERI DAL FILE
190 GET #2,G:A(E)=G
200 IF G=0 THEN GOTO 230
210 PRINT "BYTE #";E;"=";G
220 NEXT E
230 CLOSE #2
```

Figura 5-10 Esempio di programma per l'istruzione GET

Dopo aver battuto il programma-tipo PUT-GET, battete **RUN RETURN**.

Quando eseguite il programma in fig. 5-10 esso stamperà i numeri inseriti da tastiera insieme ai numeri di byte nei quali erano memorizzati.

Dopo aver battuto il programma, battete **RUN RETURN** inserendo i numeri 2, 5, 67, 54, 68.

La figura 5-11 è un esempio di esecuzione del programma PUT/GET.

BYTE #1 = 2
BYTE #2 = 5
BYTE #3 = 67
BYTE #4 = 54
BYTE #5 = 68

Figura 5-11 Esempio di esecuzione del programma PUT/GET

USO DELL'ISTRUZIONE STATUS

STATUS (ST.)

Formato: STATUS #iocb, avar
Esempio: 100 STATUS #5, ERROR

L'istruzione STATUS è usata per determinare le condizioni (stato) di un file. Questa è un'istruzione CIO e controlla diversi tipi di errori. Il primo set di errori possibili sono:

Buffer di settore disponibile?	Se no, allora ERROR-161
Numero di periferica corretto?	Se no, allora ERROR-20
Filename corretto?	Se no, allora ERROR-170
File sul dischetto?	Se no, allora ERROR-170
File bloccato? (Locked)	Se si, allora ERROR-167

Potete anche identificare tutti gli errori sui bus seriali I/O con un'istruzione STATUS. Questi sono:

Periferica fuori tempo	ERROR-138
Periferica non riconosciuta	ERROR-139
Errore sul bus seriale	ERROR-140
Dati in eccesso	ERROR-141
Errore del cheksun su bus	ERROR-142
La periferica non esegue	ERROR-144

Per usare questa istruzione, dovete aprire un file e solo input (input-only), quindi chiuderlo. Solo allora potete usare l'istruzione STATUS. È consigliabile usare la forma XIO per questa istruzione, perché è più affidabile e perché potete associare uno specifico filename all'errore che cercate di scoprire.

La figura 5-12 vi permette di controllare lo stato del vostro drive con l'istruzione TRAP. Prima di eseguire questo programma, spegnete la vostra unità a dischi.

```
10 GRAPHICS 0:REM TRAP/STATUS DEMO
20 DIM A(50),A$(10),D$(1)
30 GRAPHICS 0:? "ESEMPIO DI PROGRAMMA PUT/GET SU DISCO":?
40 ? "Devi LEGGERE o SCRIVERE ?":INPUT A$:?
50 IF A$="LEGGERE" THEN 160
60 IF A$<>"SCRIVERE" THEN PRINT "?":GOTO 40
70 REM ROUTINE DI SCRITTURA
80 TRAP 400:OPEN #1,8,0,"D1:ESEMPIO1.DAT"
90 ? "Inserisci un numero minore di 256":INPUT X
100 PUT #1,X
110 IF X=0 THEN CLOSE #1:GOTO 130
120 GOTO 90
130 GRAPHICS 0:? :? "Adesso leggo i dati dal file?":INPUT A$:?
140 IF A$="NO" THEN END
150 IF A$<>"SI" THEN 130
160 REM ROUTINE DI LETTURA
170 TRAP 400:OPEN #1,4,0,"D1:ESEMPIO1.DAT"
```

```
180 FOR E=1 TO 50
190 GET #1,G:A(E)=G
200 IF G=0 THEN GOTO 230
210 PRINT "BYTE #";E;"=";G
220 NEXT E
230 CLOSE #1
240 END
400 TRAP 40000:STATUS #1,ST:IF ST<>138 AND ST<>139 THEN PRINT "AIUTO":? ST:GOTO
410 ? "E' accesa la vostra unita' a dischi?"
420 ? "Battete Y se avete accesa la unita' a dischi":INPUT D$
430 CLOSE #1:GOTO 40
```

Figura 5-12 Esempio di programma per l'istruzione STATUS

SOSTITUZIONE DELL'ISTRUZIONE XIO PER LE OPZIONI DEL MENU DOS

XIO (X.)

Formato: XIO cmdno, #iocb, aexp1, aexp2, filespec
Esempio: 100 XIO 3, #6, 4, 0, "D: TEST.BAS"

XIO è una istruzione generale di INPUT/OUTPUT usata per operazioni speciali. Essa viene usata quando volete effettuare qualcuna delle funzioni che sarebbero altrimenti effettuate usando le operazioni del Menu DOS. Queste istruzioni XIO sono usate per aprire un file, leggere o scrivere un record o dei caratteri, chiudere un file, memorizzare lo status, riferirsi ad una locazione in un file per leggere o scrivere, cambiare nome, cancellare, bloccare o sbloccare un file. Notate che le chiamate XIO necessitano dell'identificatore dei file (filespec).

CMDNO (Command number = numero dell'istruzione) è usata per specificare quale istruzione deve essere eseguita.

CMDNO	OPERAZIONI	ESEMPIO
3	OPEN	XIO 3, #1, 4, 0, "D: TEST.BAS"
5	GET di un record	XIO 5, #1, 0, 0, "D: TEST.BAS"
7	GET caratteri	XIO 7, #1, 0, 0, "D: TEST.BAS"
9	PUT un record	XIO 9, #1, 0, 0, "D: TEST.BAS"
11	PUT caratteri	XIO 11, #1, 0, 0, "D: TEST.BAS"
12	CLOSE	XIO 12, #1, 0, 0, "D: TEST.BAS"
13	Richiesta di STATUS	XIO 13, #1, 0, 0, "D: TEST.BAS"
32	RENAME	XIO 32, #1, 0, 0, "D: OLD,NEW"
33	DELETE	XIO 33, #1, 0, 0, "D: TEMP.BAS"
35	LOCK FILE	XIO 35, #1, 0, 0, "D: ATARI.BAS"
36	UNLOCK FILE	XIO 36, #1, 0, 0, "D: DOSEX.BAS"

Nota: non usate due volte il *nome della periferica* quando cambiate nome ad un file; cioè non fate "D: OLD, D: NEW".

PROGRAMMA TIPO PER L'USO DELL'ISTRUZIONE XIO

La figura 5-13 vi permette di creare un file per ogni mese dell'anno nel quale potete inserire il nome e la data del compleanno di amici e familiari. Il programma usa l'istruzione XIO per creare un file per ogni mese, per bloccare e sbloccare ogni file quando è necessario al programma e per chiudere il file quando avete finito con esso.

La linea 20 definisce il file su disco D: BIRTHDAY come FILE$, quindi alla linea 170, FILE$ è aperto con l'istruzione XIO per l'input. L'istruzione XIO alla linea 390 sblocca il file giusto, alla linea 400 l'XIO crea il file e vi permette di scrivervi. L'istruzione XIO successiva, alla linea 430, chiude il file e l'istruzione XIO alla linea seguente blocca il file per prevenire una cancellazione accidentale.

```
5 GRAPHICS 0
10 DIM A$(5),D$(15),FILE$(20),DATE$(20),MON$(20),ERR$(20),NOME$(40)
20 FILE$="D:COMPLEAN.NO"
30 ERR$="ERRORE NEL MESE #"
100 GRAPHICS 0:? "BATTETE IL NUMERO DEL MESE (1-12)"
110 TRAP 100:INPUT MESE
120 TSTEND=0
130 MESE=INT(MESE)
140 IF MESE<1 OR MESE>12 THEN ? ERR$:GOTO 100
145 GOSUB 1000+MESE
150 FILE$(12)=STR$(MESE)
160 EOF=0
170 TRAP 700:XIO 3,#2,4,0,FILE$
180 TRAP 600:FOR I=0 TO 1 STEP 0
190 INPUT #2;NOME$
200 INPUT #2;DATE$
210 EOF=EOF+1
220 IF EOF=1 THEN ? "I COMPLEANNI IN ";MON$;" SONO:":?
224 TEMP=LEN(NOME$)
225 NOME$(TEMP+1)=" "
226 NOME$(30)=" "
227 NOME$(TEMP+2,30)=NOME$(TEMP+1)
230 ? NOME$,DATE$
240 NEXT I
299 REM NUOVI INSERIMENTI NEI COMPL.
300 ? "VUOI INSERIRE UN ALTRO COMPLEANNO?":INPUT A$
310 IF A$<>"SI" THEN GOTO 20
320 ? "BATTI IL NOME DELLA PERSONA"
330 INPUT NOME$
340 ? "BATTI LA RELATIVA DATA DI NASCITA (MM-GG-AA)"
350 INPUT DATE$
360 MESE=INT(VAL(DATE$))
370 IF MESE<1 OR MESE>12 THEN ? ERR$,DATE$:GOTO 300
380 FILE$(12)=STR$(MESE)
```

```
390 TRAP 400:XIO 36,#3,0,0,FILE$:OPEN #2,9,0,FILE$:GOTO 410
400 CLOSE #2:XIO 3,#2,8,0,FILE$
410 PRINT #2,NOME$
420 PRINT #2,DATE$
430 XIO 12,#2,0,0,FILE$
440 XIO 35,#2,0,0,FILE$
450 GOTO 300
600 CLOSE #2:IF EOF=0 THEN ? "NIENTE COMPLEANNI IN ";MON$
610 MESE=MESE+1
620 IF MESE>12 THEN MESE=1
630 TSTEND=TSTEND+1
640 IF TSTEND=1 THEN GOTO 145
650 GOTO 300
700 EOF=0:GOTO 600
1001 MON$="GENNAIO":RETURN
1002 MON$="FEBBRAIO":RETURN
1003 MON$="MARZO":RETURN
1004 MON$="APRILE":RETURN
1005 MON$="MAGGIO":RETURN
1006 MON$="GIUGNO":RETURN
1007 MON$="LUGLIO":RETURN
1008 MON$="AGOSTO":RETURN
1009 MON$="SETTEMBRE":RETURN
1010 MON$="OTTOBRE":RETURN
1011 MON$="NOVEMBRE":RETURN
1012 MON$="DICEMBRE":RETURN
3000 FILE$(12,12+LEN(STR$(MESE)))=STR$(MESE):? FILE$
```

Figura 5-13 Programma tipo per l'istruzione XIO

Quando eseguite questo programma, inserite un numero da 1 a 12. Il programma controllerà per vedere se ci sono o meno dati nel file, se non ve ne sono apparirà il messaggio: NIENTE COMPLEANNI (A) seguito dal mese selezionato. Se vi sono dati, lo schermo mostra il nome delle persone e il loro compleanno per quel mese. In entrambi i casi, lo schermo mostrerà il nome e i compleanni del mese successivo (così che non vi dimentichiate un compleanno importante che capita nella prima parte del mese successivo). Quando non desiderate vedere un altro file o fare un altro inserimento, battete **NO** ad ogni messaggio e il programma terminerà.

COME SALVARE E CARICARE PROGRAMMI E DATI CON IL BASIC ATARI

Non potete modificare un programma nella forma "tokenizzata". Quindi, per modificare un programma che è stato salvato nella forma tokenizzata, il programma dovrebbe prima essere modificato nella forma non tokenizzata. Questo evita l'incremento non necessario delle dimensioni della tabella interna dei simboli (vedere file tokenizzati e non tokenizzati). Al capitolo 3 avete usato le istruzioni SAVE e LOAD per memorizzare e riprendere il programma nella forma tokenizzata. In questo stesso capitolo, avete usato le istruzioni SAVE e LOAD per memorizzare un programma su dischetto col nome D: INTEREST.SAV. Quello che farete negli esempi che seguono, sarà di modificare INTEREST.SAV caricando prima il programma tokenizzato e quindi listarlo di nuovo sul dischetto nella forma non tokenizzata. La versione non tokenizzata potrà allora essere caricata nel computer e modificata.

Se avevate cancellato il programma INTEREST.SAV dal dischetto, rifatevi alla figura 5-14 e riscrivetelo nel computer.

```
100 REM *** INTERESSI
110 PRINT "SE BATTETE L' AMMONTARE DEL CAPITALE"
120 PRINT "INIZIALE E IL TASSO DI INTERESSE ANNUO";
130 PRINT "IO MOSTRERO' COME I VOSTRI SOLDI AUMEN"
140 PRINT "TANO ANNO DOPO ANNO, PER FERMARMI PRE"
150 PRINT "MI IL TASTO BREAK."
155 PRINT
160 PRINT "PRINCIPALE"
165 INPUT P
170 PRINT "TASSO DI INTERESSE"
175 INPUT R
180 LET N=1
190 PRINT
200 LET A=P*(1+R/100)^N
210 PRINT "ANNO=";N
220 PRINT "AMMONTO=";A
230 LET N=N+1
240 GOTO 190
```

Figura 5-14 Programma tipo: Interessi

LIST E ENTER

Qui di seguito trovate i passaggi per memorizzare e riprendere questo programma sul dischetto usando le istruzioni LIST e ENTER. Eseguite il primo set di istruzioni se avete una unità a dischi ATARI 810 e il secondo set se avete un'unità doppia a dischi ATARI 815.

Per un'unità a dischi ATARI 810

1. Usate il dischetto di sistema per caricare DOS II.

2. Togliete il dischetto di sistema e inserite il dischetto dei dati.

3. Battete **LOAD "D: INTEREST.SAV" RETURN**.

4. Battete **LIST RETURN** per vedere D: INTEREST.SAV sullo schermo.

5. Ora battete **LIST "D: INTEREST.LIS" RETURN** per memorizzare il programma non tokenizzato sul dischetto. (Il programma non apparirà sullo schermo).

6. Battete **NEW RETURN**. (Questo elimina il programma INTEREST.SAV dalla memoria).

7. Battete **ENTER "D: INTEREST.LIS" RETURN** (Questo richiama in memoria il programma nella forma non tokenizzata).

8. Battete **LIST RETURN**. Usando i tasti opportuni cambiate la linea 160 da "Principale" ad "Ammontare Principale".

9. Battete **LIST "D: INTEREST.LIS" RETURN** per memorizzare il cambiamento nel programma già sul dischetto.

10. Battete **ENTER "D: INTEREST.LIS" RETURN**.

Per una unità doppia a dischi ATARI 815:

1. Usate il dischetto di sistema nel Drive 1 per caricare DOS II e ponete il dischetto di dati nel Drive 2.

2. Battete **LOAD "D2: INTEREST.SAV" RETURN**.

3. Battete **LIST RETURN** per vedere D2: INTEREST.SAV sullo schermo.

4. Ora battete **LIST "D2: INTEREST.LIS" RETURN** per memorizzare il programma non tokenizzato sul dischetto (Il programma non apparirà sullo schermo).

5. Battete **NEW RETURN**.

6. Battete **ENTER "D2: INTEREST.LIS" RETURN**

7. Usate i tasti opportuni per cambiare la linea 160 da "Principale" ad "Ammontare Principale".

8. Battete **LIST "D2: INTEREST.LIS" RETURN** per memorizzare il cambiamento nel programma già sul dischetto.

9. Battete **ENTER "D2: INTEREST. LIST" RETURN**.

Per caricare e far partire il programma dal dischetto:

Battete **RUN RETURN**. Il computer riprende ed esegue il programma (vedere la figura 5-15). I numeri inseriti per l'istruzione INPUT erano 1200 dollari per il capitale e 12% per l'interesse.

RUN
SE BATTI L'AMMONTARE DEL CAPITALE E IL TASSO D'INTERESSE ANNUALE IO MOSTRERÒ COME I TUOI SOLDI CRESCONO ANNO DOPO ANNO,

PER FERMARMI PREMI BREAK

```
CAPITALE?1200
TASSO DI INTERESSE?12

ANNO = 1
AMMONTARE = 1343.999988

ANNO = 2
AMMONTARE = 1505.28

ANNO = 3
AMMONTARE = 1685.913576

STOPPED AT LINE 200
```

Figura 5-15 Esempio di esecuzione del programma Interest

Se desiderate salvare il programma originale E la versione modificata, date un nuovo nome al programma modificato.

Per fare cambiamenti nel programma eseguite queste istruzioni:

Per salvare la versione listata sul dischetto:

Battete **LIST "D: INTEREST.LIS" RETURN** e quelle linee specifiche saranno sostituite nel programma esistente.

Per richiamare la versione listata dal dischetto:

1. Battete **NEW RETURN** (Questo cancella dalla memoria la versione tokenizzata e la sua tabella dei simboli).

2. Battete **ENTER "D: INTEREST" RETURN** e riporterete la versione non tokenizzata in memoria.

3. Battete **LIST RETURN**.

Nota: cambiate il nome al programma modificato se volete conservare sia il programma originale che quello modificato. Potete usare LIST ed ENTER anche per memorizzare e richiamare file di dati. Un file di dati non contiene istruzioni; contiene nomi, indirizzi, ammontare di assegni, etc. Per accedere a questi file avete bisogno di un programma. Così prima di continuare, battete il programma di dati in figura 5-16. Quando volete fermare l'esecuzione del programma, battete zero per l'inserimento del successivo numero di assegno.

```
1 REM QUESTO PROGRAMMA SCRIVE UN FILE DI NUMERI DI ASSEGNI E IL LORO AMMONTARE
5 DIM CHECKNAME$(40)
7 REM ** OPEN CON 8 CREA UN FILE DI DATI **
10 OPEN #1,8,0,"D:CHECKS"
20 CHECKAMT=0:CHECKNAME$=" "
25 PRINT "NUMERO ASSEGNO";
30 INPUT CHECKNUM
35 IF CHECKNUM=0 THEN 80
38 REM SCRIVE IL DATO SUL DISCHETTO
```

```
40 PRINT "AMMONTARE DELL' ASSEGNO";
45 REM CHIUDE IL FILE DI DATI
50 INPUT CHECKAMT
55 REM APRE IL FILE DI DATI PER LA
58 REM LETTURA.OPEN CON 4=SOLO LETTURA
60 PRINT "A CHI ERA INTESTATO L' ASSEGNO"
65 REM LEGGE IL DATO DAL DISCHETTO
70 INPUT CHECKNAME$
75 REM SCRIVE IL DATO
80 PRINT #1:IF CHECKNUM>0 THEN PRINT :GOTO 20
85 REM ** CHIUDE IL FILE DI DATI **
90 CLOSE #1
100 CLOSE #1
```

Figura 5-16 Programma tipo per creare un file di Dati

OPEN E CLOSE

Per accedere al file di dati, D: CHECKS, viene usata l'istruzione BASIC OPEN. Se non ci sono file con quel nome sul dischetto, il file viene creato automaticamente. Per memorizzare e riprendere un dato da un programma sul dischetto:

1. Accendete l'unità a dischi.
2. Mettete il dischetto di sistema nell'unità.
3. Accendete il computer e inizializzate (Boot) il sistema.
4. Battete OPEN #1, 8, 0, "D: DATA" **RETURN**, questo informa il computer di permettere la scrittura di DATI nel file sul dischetto nel Drive 1.
5. Battete PRINT #1; X; ","; Y; ","; Z **RETURN** (X, Y, Z sono numeri).
6. Battete CLOSE #1 **RETURN** che informa il computer che il file è terminato.

La figura 5-17 è un esempio di esecuzione del programma in fig. 5-16. Abbiamo usato i numeri 100, 101, 102 come NUMERI DI ASSEGNO; 12,50; 24,35; 102,67 per AMMONTARE; John Smith, George Brown e Heavy per: CHI È STATO CONTROLLATO?

READY
RUN
NUMERO DI ASSEGNO?100
AMMONTARE ASSEGNO?12.50
A CHI È STATO INTESTATO?JOHN SMITH

NUMERO DI ASSEGNO?101
AMMONTARE ASSEGNO?24.35
A CHI È STATO INTESTATO?GEORGE BROWN

NUMERO DI ASSEGNO?102
AMMONTARE ASSEGNO?102.67
A CHI È STATO INTESTATO?HEAVY

NUMERO DI ASSEGNO?0

Figura 5-17 Esecuzione del programma tipo per dati

La figura 5-18 mostra come sono memorizzate sul dischetto le informazioni in figura 5-17.

100, 12.51, JOHN SMITH
101, 24.35, GEORGE BROWN
102, 102.67, HEAVY
0, 0

Figura 5-18 Le informazioni memorizzate sul dischetto

ACCESSO AD UN FILE DANNEGGIATO

Ci sono due motivi per cui un file può risultare danneggiato:

1. La registrazione nella lista dei file (disk directory) che contiene il nome del file, il puntatore al primo settore, il numero di settori, non è leggibile.

2. Il file medesimo è rovinato.

Potrebbe essere danneggiata la registrazione nella DISK DIRECTORY, nel qual caso non è possibile accedere al file. Se è stata cancellata accidentalmente, apparirà sullo schermo un ERROR-170, File not found (file non trovato). Se il numero di settori indicato nella DISK DIRECTORY non coincide col numero di settori effettivamente occupati dal file, apparirà sullo schermo un ERROR-164 (File Number Mismatch), in quest'ultimo caso, potete recuperare la porzione del file che occupa i settori registrati nella Directory eseguendo il programma Get Byte in figura 5-19.

```
10 OPEN #1,4,0,"D:FILE.1"
20 OPEN #2,8,0,"D:FILE.2"
25 TRAP 50
30 GET #1,A
40 PUT #2,B
45 GOTO 30
50 CLOSE #1
60 CLOSE #2
```

Figura 5-19 Programma Get Byte

Chiameremo: File 1 il file danneggiato e File 2 il file recuperato.

Nota: è possibile leggere i settori che si trovano PRIMA dei settori danneggiati. Tutti gli altri settori, successivi a quello danneggiato non sono accessibili. Sarebbe meglio copiare (COPY) i file intatti dal dischetto danneggiato su un nuovo dischetto, per evitare ulteriori problemi.

Se il file stesso è danneggiato, potete ancora usare il programma Get Byte che trasferirà ogni settore buono dal file danneggiato in un altro file.

IL FILE AUTORUN.SYS

Quando esiste un file AUTORUN.SYS nel Drive 1, quel file sarà automaticamente caricato in RAM ed eseguito ogni volta che accendete il sistema. L'intero processo viene completato prima che il controllo del sistema torni all'utente. Il file AUTORUN.SYS può contenere dati; può essere anche codice oggetto che viene caricato ma non eseguito oppure codice oggetto che viene caricato ed eseguito immediatamente.

La figura 5-20 illustra l'uso dell'AUTORUN.SYS per caricare direttamente il DOS anche se è presente un cartridge. Dopo l'esecuzione, normalmente AUTORUN-.SYS restituisce il controllo alla routine di inizializzazione del DOS. Comunque l'inizializzazione del sistema deve essere completata prima di procedere. Questo si ottiene modificando due locazioni di memoria del sistema operativo: COLDST all'indirizzo 244 (hex) e BOOT all'indirizzo 9 (hex). COLDST deve essere messo a 0 e BOOT a 1.

Il programma listato di seguito, mette queste due locazioni al loro valore e quindi salta direttamente al vettore d'inizio DOS.

Se non avete un cartridge Assembler Editor, potete creare un file equivalente usando l'istruzione BASIC POKE e quindi salvando il Binary File nel DOS. La lista dei numeri decimali da inserire è la seguente:

Indirizzo decimale	**Codici decimali**
15000	162, 00,
15002	142, 68, 02,
15005	232,
15006	134, 09
15008	108, 10, 00

Quando sono stati inseriti questi codici in BASIC, battete **DOS RETURN** per salvare il file usando l'opzione del Menu DOS: K. BINARY SAVE.

Notate che non sono stati inseriti i numeri per il parametro INIT. Se spegnete il vostro computer e quindi lo riaccendete, dovreste entrare direttamente nel DOS. Per tornare al BASIC battete semplicemente **B RETURN** o premete **SYSTEM RESET**.

```
; Programma Autorun
;
; Lancia DOS senza passare dal cartridge.
;
COLDST = $244
BOOT = $09
DOSVEC = $0A
        * = 3A98

                                  (CODICE HEX)
DOSGO   LDX #0                    A2 00
        STX COLDST                8E 44 02
        INX                       E8
        STX BOOT                  86 09
        MP (DOSVEC)               6C 0A 00
        * = $2E0                  indirizzo d'inizio a 2E0
        . WORD DOSGO              98 3A
        . END
```

SELECT ITEM OR RETURN FOR MENU
K **RETURN**
SAVE-GIVE FILE, START, END [,INIT,RUN]
AUTORUN.SYS, 3A98, 3AA2,, 3A98
SELECT ITEM OR RETURN FOR MENU

Figura 5-20 Una applicazione avanzata del programma AUTORUN. SYS

6

INFORMAZIONI SUL SISTEMA UNITÀ A DISCO

DISCHETTI ATARI

I dischetti ATARI sono lamine circolari e sottili coperte da un ossido simile a quello usato sulle audio cassette. Ogni dischetto ATARI ha un diametro di 5 1/4 pollici [13 cm ca. N.D.T.] ed è sigillato in una speciale custodia nera, progettata per proteggerlo dalle deformazioni, dalle rotture o da danneggiamenti in genere.

Ogni Unità a dischi richiede, per poter funzionare, il tipo di dischetto progettato specificamente per essa. L'Unità a dischi ATARI 810 è un'unità a dischi a densità singola e l'Unità doppia a dischi ATARI 815 è un'unità a dischi a doppia densità. La differenza essenziale fra i dischetti a doppia e quelli a singola densità è che la tecnica di registrazione in doppia densità richiede una superficie di registrazione di alta qualità, che consente di memorizzare il doppio di dati nello stesso spazio. Entrambi i tipi di dischetto sono fatti allo stesso modo; la differenza è che, dai dischetti di densità doppia, viene pretesa la garanzia che lavoreranno con la tecnica di registrazione in doppia densità. Un dischetto a doppia densità vuoto può essere usato anche su una unità a dischi a singola densità ma non è possibile usare un dischetto a singola densità su un'unità a doppia densità.

Una volta che un dischetto sia stato formattato per essere usato in una unità a dischi in singola densità, non è più possibile utilizzarlo in una unità a doppia densità, salvo riformattarlo in una unità a doppia densità; è anche vero il contrario. Se avete un sistema che usi entrambi i tipi di drive e di dischetti, etichettate chiaramente il dischetto affinché vi si legga su quale tipo di drive è stato formattato.

UNITÀ A DISCHI ATARI 810

L'unità a dischi ATARI 810 è un singolo drive che registra in singola densità. Usa dischetti flessibili standard da 5 1/4 pollici: ATARI 810 Master Diskette II (CX8104) (dischetto principale II ATARI 810), ATARI 810 Blank Diskettes (CX8100, dischetti blank ATARI 810), ATARI 810/815 Blank Diskettes (CX8202) e ATARI 810 Formatted Dikettes II (CX 8111, Dischetti Formattati II ATARI 810) ognuno dei quali memorizza 88 K (88 mila) bytes. L'unità a dischi ATARI 810 incorpora un microprocessore che le conferisce capacità autonoma. Questo signfica che il motore del drive non è costantemente in funzione, ma aspetta che "gli si dica" quando necessario.

FUNZIONI DELL'UNITÀ A DISCHI

Quando inserite un dischetto, il foro al centro è posizionato automaticamente sull'albero del drive e il dischetto è fissato. Il dischetto circolare ruota nella sua custodia protettiva. Quando accedete al dischetto, la testina magnetica è posizionata sulla superficie di lettura/scrittura.

Quando memorizzate un dato su un dischetto, il drive converte i dati che riceve dalla console del computer in impulsi elettrici codificati. Questi impulsi magnetizzano minuscole aree sulla superficie di ossido sul dischetto mentre questi gira.

Quando richiamate dati da un dischetto, il drive posiziona la testa magnetica cosicché l'area del dischetto dove il dato è memorizzato passi sotto di esso. Il microprocessore del drive controlla il posizionamento e la temporizzazione del dischetto.

APPENDICE A

INDICE ALFABETICO DI PAROLE RISERVATE DEL BASIC USATE PER LE OPERAZIONI CON DISCHI

Nota: il punto è obbligatorio dopo tutte le istruzioni abbreviate.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLA ISTRUZIONE BASIC
CLOSE	CL.	Istruzione I/O usata per chiudere un file su disco al termine di una operazione I/O.
DOS	DO.	Questa istruzione fa apparire il Menu DOS. Il Menu contiene tutte le opzioni. Passa il controllo dal cartridge alle utility DOS.
END		Ferma l'esecuzione di un programma, chiude i file ed elimina i suoni. Il programma può essere fatto ripartire con CONT. (**Nota:** END può essere usato più di una volta in un programma).
ENTER	E.	Istruzione I/O usata per richiamare un programma listato nella sua forma non tokenizzata (testuale). Se un programma o alcune linee sono caricate in RAM quando vi risiede un altro programma, ENTER unirà (merge) i due programmi. Se non volete la fusione battete NEW prima di usare ENTER per caricare un programma in RAM.
GET	GE.	Usata con i dischi per immettere un singolo byte di dati in una variabile da una periferica.
INPUT	I.	Questa istruzione richiede dati da una periferica. Quando questa non è specificata, si sottintende E: (editor dello schermo).
LIST	L.	Questa istruzione invia la versione non tokenizzata di un programma ad una periferica.
LOAD	LO.	Istruzioni di I/O usata per richiamare un programma salvato nella forma tokenizzata da una periferica.
NOTE	NO.	Questa istruzione, memorizza il numero assoluto del settore su disco e il numero del byte del puntatore del file (file pointer) in due variabili aritmetiche.
OPEN	O.	Apre il file specifico per operazioni di input output. Determina il tipo di operazioni permesse su un file.

POINT	P.	Questa istruzione è usata per mettere il puntatore del file ad una specifica locazione (settore e byte) sul dischetto.
PRINT	PR. o ?	Istruzione di I/O: invia l'output dal computer ad una periferica in formato record.
PUT	PU.	Causa l'output di un singolo byte di dati, cioè un carattere, dal computer ad una periferica.
RUN	R.	Carica ed inizia l'esecuzione di un determinato file.
SAVE	S.	Istruzioni di I/O: registra la versione tokenizzata di un programma in un file su una periferica.
STATUS	ST.	Chiama la routine di stato per una periferica.
TRAP	T.	Dirige l'esecuzione ad una linea specifica in caso di errore nel programma, permettendo all'utente di mantenere il controllo del programma e di correggere gli errori.
XIO	X.	Istruzione di I/O generale: usata in un programma per eseguire una istruzione del Menu DOS e specifiche istruzioni I/O.

APPENDICE B

NOTAZIONI E TERMINOLOGIA USATI COL DOS II

SYSTEM RESET	Premere il tasto **SYSTEM RESET** sulla tastiera.
RETURN	Premere il tasto **RETURN** sulla tastiera.
[ ]	Parentesi. Le parentesi racchiudono informazioni (item) opzionali.
	Puntini. I puntini che segnano una informazione racchiusa in parentesi indicano che potete ripetere tale informazione (item) quante volte volete ma non è indispensabile farlo.
{ }	Parantesi grafe. Se vi sono diversi elementi in fila verticale in parentesi grafe significa che potete scegliere quale di questi inserire. Sceglietene solo uno da porre nella vostra istruzione.
LETTERE MAIUSCOLE	Le lettere maiuscole indicano istruzioni o altro che dovete battere esattamente come appaiono.
. ,/: ; "	Punteggiatura. Questa punteggiatura deve essere battuta come mostrato nel formato dell'istruzione. Non battete parentesi quadre ne grafe.
cmdno	Numero dell'istruzione. Da usarsi con l'istruzione XIO.
exp	Espressione. In questo manuale, le espressioni sono divise in tre tipi: aritmetiche, logiche e stringhe.
aexp	Espressione aritmetica. Composta genericamente di variabili, funzioni, costanti o due espressioni separate da un operatore aritmetico (aop).
aexp1	Espressione aritmetica 1. Questa espressione aritmetica rappresenta il primo byte ausiliario di controllo I/O quando usato in una istruzione del tipo OPEN.
aexp2	Espressione aritmetica 2. Questa espressione rappresenta il secondo byte ausiliario di controllo I/O quando usato in una istruzione del tipo OPEN. Generalmente uguale a 0.
Identificatore di file (filespec)	Generalmente una espressione stringa che si riferisce ad un file e alla periferica nel quale esso si trova ad esempio: "D1:MYPROG.BAS" per un file nel Drive 1.
IOCB	Blocco di controllo Input/Output. Una espressione aritmetica con valore da 1 a 7. L'IOCB è usato per riferirsi ad una periferica o file. IOCB 0 è riservato al BASIC per lo Screen Editor e può essere usato dall'utente solo se lo Screen Editor non viene usato.
lineno	Numero di linea. Una costante che identifica una linea di programma particolare in un programma BASIC in modo differito. Un numero di linea può essere un numero intero da 0 a 32767. I numeri di linea determinano l'ordine dell'esecuzione di un programma.

var	Variabile di qualsiasi tipo. In questo manuale le variabili sono classificae in aritmetiche (avar), matrici (mvar) o stringhe (svar).
avar	Variabile aritmetica. Una locazione nella quale un valore numerico è memorizzato. Un nome di variabile può essere costituito di un certo numero (da 1 a 120) di caratteri alfanumerici, ma deve iniziare con un carattere alfabetico maiuscolo non in reverse.
mvar	Matrice, chiamata anche variabile con indice (subscipted). Un elemento di un array o matrice.
svar	Variabile stringa. Una locazione nella quale può essere memorizzata una stringa di caratteri.

APPENDICE C

MESSAGGI D'ERRORE BASIC E RIMEDI

Nota: i messaggi d'errore da 2 a 21 possono verificarsi solo nell'esecuzione di un programma BASIC.

N. ERRORE	NOME ERRORE	CAUSA E RIMEDIO
2	Memoria insufficiente	Non avete abbastanza memoria per memorizzare la linea o per dimensionare una nuova variabile stringa. Cancellate una variabile non usata o aggiungete memoria (vedere Capitolo 11 del manuale BASIC, per la conservazione di memoria).
3	Errore nel valore	O l'intero positivo atteso era negativo o il valore non era nei limiti attesi.
4	Troppe variabili	Avete superato il valore massimo (128) di nomi di variabili e dovete cancellarne alcune. (Vedere capitolo 11 del Manuale BASIC).
5	Errore nella lunghezza di una stringa	Avete tentato di leggere o scrivere in una locazione al di fuori delle dimensioni della stringa o avete usato zero come indice. Ampliate la lunghezza assegnata con la DIM e non usate zero come indice.
6	Fuori dai DATA	Non avete abbastanza dati nelle istruzioni DATA per le istruzioni READ.
7	Numero di linea maggiore di 32767	Controllate i rimandi ai numeri di linea nelle istruzioni del tipo GOTO e RESTORE.
8	Errore nella istruzione input	Avete tentato di assegnare un valore non numerico ad una variabile numerica. Controllate il tipo di variabili e/o i dati in input.
9	Errore nel DIM	Le dimensioni del DIM eccedono 5460 per array numerici o 32767 per stringhe; un array o una stringa è stato ridimensionato; è stato fatto riferimento ad un array o stringa non dimensionati.
11	Overflow/ Underflow in virgola mobile	Avete tentato di dividere per zero o di riferirvi ad un numero con valore assoluto minore di 1E-99 o maggiore o uguale a 1E-98.
12	Linea non trovata	Un GOTO, GOSUB o THEN riferito ad una linea che non esiste.

13	NEXT senza FOR	Un'istruzione NEXT è stata incontrata senza che vi fosse il FOR corrispondente.
14	Linea troppo lunga	Avete ecceduto la lunghezza del buffer per l'esecuzione nella linea BASIC.
15	Linee GOSUB o FOR cancellate	Una istruzione NEXT o RETURN è stata incontrata e il corrispondente FOR o GOSUB è stato cancellato dall'ultima volta che il programma è stato eseguito.
16	Errore sul RETURN	Controllate il programma per il GOSUB mancante.
17	Errore di sintassi	Il computer ha incontrato una linea con errore di sintassi, correggete tale linea.
18	Errore nella funzione VAL	La stringa nella funzione VAL non era numerica.
19	LOAD di un programma troppo lungo	Non avete abbastanza memoria per caricare il vostro programma.
20	Errore nel numero della periferica	Avete inserito un numero di periferica non compreso fra 1 e 7.
21	Errore nel LOAD di un file	Avete cercato di caricare un file che non ammette l'istruzione LOAD, non era un file BASIC tokenizzato; questi file vengono creati con l'istruzione SAVE.

Nota: i seguenti sono errori di input/output che si hanno durante l'uso di unità a dischi, stampati o altri dispositivi accessori. Ulteriori informazioni sono fornite con l'hardware accessorio.

128	Interruzione BREAK	L'untente ha premuto **BREAK** durante le operazioni I/O fermandone l'esecuzione.
129	IOCB* già aperto	Istruzione OPEN in un ciclo di programma o IOCB già in uso per un altro file o periferica.
130	Periferica non esistente	Avete cercato di accedere ad una periferica non definita. Questo errore si può avere tentando di accedere all'interfaccia ATARI 850 senza aver eseguito RS232-C AUTORUN.SYS. Un altro caso comune in cui si ha questo errore è quando viene specificato un nome di file senza aver indicato la periferica. Esempio: "MYFILE" invece di "D:MYFILE". Controllate e correggete l'istruzione che ha generato l'errore.
131	IOCB a sola scrittura	Avete tentato di leggere da un file a sola scrittura; aprite il file per la lettura o per le modifiche.

132	Istruzione di gestione scorretta	È un codice d'errore CIO. Il codice passato alla routine di gestione periferica è scorretto. L'istruzione è < = 2 oppure è un'istruzione speciale data ad una routine di gestione periferica (handler) che non ha implementato alcuna istruzione speciale. Controllate che l'istruzione XIO o il codice IOCB non siano scorretti.
133	File/periferica non aperta	Non avete aperto questo file o periferica. Controllate l'istruzione OPEN o l'istruzione I/O per il file.
134	Numero IOCB scorretto	Avete tentato di usare un indice IOCB scorretto. Per il BASIC, varia da 1 a 7 e non è permesso l'uso di IOCB 0. Il cartridge Assembler Editor richiede che l'indice IOCB sia un multiplo di 16 e minore di 128.
135	IOCB a sola lettura	Avete tentato di scrivere su una periferica o file aperta per sola lettura. Aprite il file per scrivere o modificare.
136	Fine del file	Il file in input è alla fine. Non vi sono più dati.
137	Record troncato	Questo errore si ha tipicamente quando il record in lettura è più lungo della lunghezza specificata nella chiamata a CIO (la lunghezza massima di un record BASIC e 119 bytes); ad esempio quando tentate di usare un INPUT (istruzione orientata a records) su un file che è stato creato con l'istruzione PUT (orientata a byte).
138	Periferica fuori tempo	Quando avete inviato un'istruzione su un bus seriale, la periferica non ha risposto nel periodo stabilito dal sistema operativo per quell'istruzione. Il numero della periferica è sbagliato o l'utente ha specificato la periferica sbagliata; la periferica non c'è; non è in grado di rispondere nel periodo corretto; oppure non è attaccata. Se la periferica è un registratore, la velocità del nastro potrebbe essere stata alterata o il nastro posizionato impropriamente. Esaminate tutti i collegamenti per assicurarvi che siano a posto; controllate l'unità a dischi per assicurarvi che sia accesa e con i codici posizionati correttamente. Controllate che la vostra istruzione abbia un numero di drive corretto. Ridate l'istruzione, se l'errore ritorna, fate controllare l'unità a dischi.
139	Periferica non riconosciuta	La periferica non può rispondere a causa di parametri scorretti come ad esempio un settore non indirizzabile. La periferica potrebbe anche

		aver ricevuto una istruzione scorretta o aver ricevuto dati sbagliati dal Computer. Correggete i parametri dell'istruzione. Controllate anche i cavi I/O. Questo è un errore specifico della periferica, quindi fate riferimento alla documentazione di quella periferica.
140	Errore negli impulsi seriali	Il bit 7 di SKSTAT del chip POKEY è messo a 1. Questo significa che le comunicazioni fra la periferica e il Computer sono alterate. È un errore molto raro ed è fatale. Se capita più di una volta, fate controllare il vostro Computer o la periferica. Potete anche rimuovere le periferiche una alla volta per isolare il problema. Per le cassette provate i rimedi suggeriti per l'errore 138.
141	Cursore fuori dai limiti	Il vostro cursore è fuori dai limiti per il modo grafico particolare che avete scelto. Cambiate i parametri del pixel.
142	Dati seriali in eccesso	Il bit 5 di SKSTAT in POKEY è messo a 1. Il computer non risponde abbastanza velocemente ad una interruzione dell'input su bus seriale o POKEY riceve una seconda parola a 8-bit prima che il computer abbia trattato la precedente. È un errore raro, se capita più di una volta fate controllare il computer.
143	Errore del checksum	Le comunicazioni col bus seriale sono alterate. Il checksum inviato dalla periferica non è uguale a quello calcolato in base all'impulso ricevuto dal computer. Non ci sono rimedi codificati perché si può trattare sia di un problema hardware che software.
144	La periferica non esegue	La periferica non è in grado di eseguire una istruzione valida. Avete tentato di scrivere su un dischetto o periferica protetta o l'unità a dischi non è in grado di leggere/scrivere al settore richiesto. Rimuovete la linguetta protettiva o spegnete l'interruttore di protezione. Vedere i manuali specifici per le altre periferiche.
145	Modo scorretto dello schermo	Avete tentato di aprire lo Screen Editor con un numero scorretto di modo grafico. Controllate la chiamata al modo grafico o il byte aux2 nell'IOCB.
146	Funzione non implementata	Il sistema di gestione periferica non contiene la funzione; si ha, ad esempio, tentando una istruzione PUT in tastiera o scrivendo un'istruzione speciale per la tastiera. Controllate che l'istruzione I/O sia corretta e che lo sia anche la periferica.

147	RAM insufficiente	Non c'è abbastanza memoria per il modo grafico scelto, aggiungete altra memoria o usate un modo grafico che non richieda tanta memoria.
160	Errore nel numero del drive	Avete specificato un numero di drive che non era fra 1 e 8, non era allocato un buffer per il drive o il drive non era stato acceso all'inizio. Fate riferimento al capitolo 1 di questo manuale. Controllate il filespec o il byte 1802 per il numero dei buffer allocati per i drive.
161	Troppi file aperti	Non avete più buffers liberi da usare per un altro file. Controllate la locazione 1801 per il numero dei buffers disponibili. Assicuratevi anche che non siano aperti file che non dovrebbero essere aperti.
162	Disco pieno	Non avete più settori liberi sul dischetto, usate un dischetto che ne abbia.
163	Errore I/O di sistema irrecuperabile	Questo significa che c'è un errore nel File Manager. Il vostro DOS sul dischetto potrebbe non essere buono. Riprovate usando un altro DOS.
164	Numero del file spaiato	La struttura del file è danneggiata o il valore POINT è sbagliato. La tabella dei file punta ad un settore appartenente ad un altro file. Spegnete il sistema e ritentate l'esecuzione del programma, se questo fallisce, avete perduto il file. Cercate di recuperare gli altri file sul dischetto, quindi riformattatelo.
165	Errore nel nome del file	La vostra specifica dei file contiene caratteri scorretti, controllatela e togliete i caratteri scorretti.
166	Errore nel dato lunghezza in POINT	Il numero del byte nella chiamata POINT era più grande di 125 (densità singola) o 253 (densità doppia). Controllate i parametri nell'istruzione POINT.
167	File bloccato	Avete tentato di accedere ad un file bloccato per scopi diversi dalla lettura. Usate l'opzione del Menu DOS G. per sbloccare il file e ritentate l'istruzione.
168	Istruzione per periferica non valida	Avete scritto un'istruzione scorretta per l'interfaccia software di una periferica. Controllate la documentazione per quella periferica e ritentate l'istruzione.
169	Lista dei file completa	Avete usato tutto lo spazio destinato alla lista dei file (Directory).

170	File non trovato	Avete tentato di accedere ad un file che non esiste nella lista dei file sul dischetto. Usate l'opzione A. del Menu DOS per controllare la corretta scrittura del nome del file e per assicurarvi che il file sia sul dischetto cui accedete.
171	POINT non valido	Avete tentato di puntare ad un byte in un file non aperto per l'aggiornamento. Controllate i parametri della istruzione OPEN o il byte aux1 della IOCB usata per aprire quel file.
172	Aggiunta scorretta	Avete tentato di aprire un file DOS I per aggiunte usando DOS II. DOS II non può fare unioni a file del DOS I. Copiate (COPY) il file DOS I in un dischetto DOS II usando il DOS II.
173	Settori danneggiati alla formattazione	L'unità a dischi ha trovato settori danneggiati mentre formattava un dischetto. Usate un altro dischetto; se questo errore si ha con più di un dischetto, la vostra unità a dischi può necessitare di riparazione.

APPENDICE D

MAPPA DI MEMORIA DOS II PER IL SISTEMA DA 32 K RAM

INDIRIZZI Decimale	Esadecimale	CONTENUTI
65535	FFFF	SISTEMA OPERATIVO
49152	C000	
49151	BFFF	CARTRIDGES
32768	8000	
32767	7FFF	AREA DELLO SCREEN DISPLAY (memoria di video)
varia		
varia		AREA DEL PROGRAMMA DELL'UTENTE — HIMEM** LOMEM*
varia		
	2S { 3305 – 1D7C } 2D { 3A38 – 247C }	S B 7, S B 6, S B 5, S B 4, S B 3 — PROGRAMMI DI UTILITÀ DEL DISCO (Disk Utility Programs)
		Buffer 1 di settore, Buffer 2 di settore, Buffer del Drive 4, Buffer del Drive 3, Buffer del Drive 2 — AREA DEI BUFFER RISERVATI AL DOS II
6781	1A7D	Buffer del Drive 1
6780	1A7C	MINI-DOS (PORZIONE DEL DUP RESIDENTE IN RAM)
5440	1540	
5439	153F	SOTTOSISTEMA PER LA GESTIONE DEI FILE
1792	0700	
1791	06FF	
	0600	RAM DISPONIBILE ALL'UTENTE
	05FF	
0	0000	RAM DEL SISTEMA OPERATIVO

* Varia col numero di buffer dei Drive e di settore.
** Dipende dal modo grafico in uso.

Nota 1: per una data configurazione di Buffer dei Drive e dei Buffer di Settore, LOMEM, può essere determinato eseguendo delle PEEK alla locazione 2E7 (Bassa) e 2E8 (Alta) esadecimali o 743 (Bassa) e 744 (Alta) decimali.

Nota 2: per determinare l'ammontare dell'area di programma disponibile all'utente, potete sia far uso della istruzione BASIC FRE(O) o eseguire delle PEEK alle locazioni 2E5 (Bassa) e 2E6 (Alta) esadecimali o 741 (Bassa) e 742 (Alta) decimali.

APPENDICE E

TABELLA DI CONVERSIONE ESADECIMALE-DECIMALE

QUATTRO CIFRE ESADECIMALI

4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	206	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

* Usate questa tabella per convertire numeri fino a quattro cifre.

Per esempio: per convertire il numero esadecimale 1234 a decimale, sommate il valore ricavato da ognuna delle quattro colonne della tabella. Per 1 usate la colonna 4 e così via.

1234 esadecimale =

```
        4096
      +  512
      +   48
      +    4
      ------
4660 decimale
```

Altri esempi:

EEDD esadecimale =

```
         57344
       +  3584
       +   208
       +    13
       -------
61149 decimale
```

AB esadecimale

```
         160
        + 11
        ----
171 decimale
```

APPENDICE F

COME ACCELERARE IL TRASFERIMENTO DI DATI ALL'UNITÀ A DISCHI

Il vostro nuovo DOS II versione 2OS ha la capacità di scrivere con lettura di verifica (una tecnica di sicurezza che dovrebbe essere usata quando è più importante l'affidabilità della velocità di trasmissione). Il dischetto principale DOS II vi viene fornito con questa capacità. Per guadagnare tempo, tuttavia, l'informazione può essere scritta sul dischetto senza la lettura di verifica. La locazione di memoria 1913 (dec.) contiene il dato che determina se debba esservi (50 hex, 80 dec.) o meno (57 hex, 87 dec.) lettura di verifica. Una scrittura senza lettura di verifica è, naturalmente, più veloce ma non può essere altrettanto affidabile. Per modificare la vostra versione del DOS II da BASIC dovete fare:

```
POKE 1913,80
```

per la scrittura veloce (senza lettura di verifica). Se volete ritornare alla scrittura con lettura di verifica:

```
POKE 1913,87
```

Per modificare la versione del DOS memorizzata sul dischetto, battete semplicemente DOS e quindi usate l'opzione "H" (WRITE DOS FILE) del Menu DOS per memorizzare la nuova versione del DOS su dischetto.

APPENDICE G

COME INCREMENTARE LO SPAZIO RAM DELL'UTENTE

Questa sezione spiega come cambiare un numero in una locazione RAM a seconda del numero di drive collegati al Personal Computer ATARI. Dovrete eseguire delle POKE nella locazione RAM 1802 (decimale) per inserirvi i codici corretti dei drive. La tabella seguente dà i valori da inserire in base al numero di drive. Notate che possono essere usati fino ad un massimo di 4 unità a dischi ATARI 810 visto che i numeri di codice vanno da 1 a 4. Notate anche che nei codici binari dei drive c'è un 1 in corrispondenza di ogni drive nel sistema. Benché in precedenza, in questo manuale, vi sia stato suggerito di assegnare i numeri 1 e 2 alla unità doppia a dischi ATARI 815 e il numero 3 all'unità a dischi ATARI 810, potete notare, dalla lista seguente, che ci sono altre opzioni.

CODICI PER NUMERO DI UNITÀ ATARI 810 ATTACCATI

Drive	Codice decimale drive	Codice esadecimale drive
Drive 1	01	00000001
Drive 2	02	00000010
Drive 3	04	00000100
Drive 4	08	00001000
Drives 1 + 2	03	00000011 (mancante)
Drives 1 + 3	05	00000101
Drives 1 + 2 + 3 + 4	15	00001111

Se il vostro sistema include sia l'unità a dischi ATARI 810 che l'ATARI 815, dovrete usare l'unità a dischi ATARI 810 in posizione da 1 a 4. L'unità doppia a dischi può essere posizionata in uno dei seguenti modi: (1 + 2), (3 + 4), (5 + 6) o (7 + 8). La tabella seguente dà i numeri da inserire dato il numero di unità a dischi doppie. Il dato è sempre memorizzato alla locazione RAM 1802 (decimale).

Drives 1 + 2	03	00000011
Drives 3 + 4	12	00001100
Drives 5 + 6	48	00110000
Drives 7 + 8	192	11000000

Esempio 1.

Se il vostro sistema include una unità a dischi ATARI 810 e una unità a dischi doppia ATARI 815, come strutturare il vostro DOS? Visto che dovete avere un Drive 1, potete considerare l'ATARI 815 come Drive 1 e 2 e l'ATARI 810 come Drive 3, oppure l'ATARI 810 come Drive 1 e l'ATARI 815 come 3 e 4. Supponendo che scegliate la prima ipotesi — l'ATARI 815 come 1 e 2 — dovete usare il codice 03 (vedi tabella più sopra). Assegnando all'ATARI 810 la posizione 3, avrete un codice 04. Il valore da porre alla locazione RAM 1802 sarà: 03 + 04 = 07 in decimale, cioè 00000111 in binario.

Esempio 2.

Supponete di avere tre unità a dischi ATARI 810 e due unità a dischi doppia ATARI 815. Siccome per l'ATARI 810 si possono usare solo le posizioni da 1 a 4, potete ricavare dalla tabella più sopra i codici 01, 02, 03. Ponendo le due unità doppie a dischi alle posizione 5 + 6 e 7 + 8 otterrete i codici 48 e 192. Sommate i valori per ottenere il valore totale: cioè 01 + 02 + 03 + 48 + 192 = 246. Questo valore 246 è il valore da porre nella locazione RAM 1802. 246 corrisponde a 11110110.

APPENDICE H

DIFFERENZE PRINCIPALI FRA DOS I (24/9/79) E DOS II

DOS I	DOS II
Il tempo d'"inizializzazione" (Boot time) è di 11 secondi.	Il tempo d'"inizializzazione" è di 7 secondi.
Il tempo di caricamento dell'insieme di utility su disco (Disk Utility Package) è nullo poiché venivano caricate durante l'inizializzazione.	Il tempo di caricamento del DUP è di 9 secondi.
Non erano ammessi caratteri jolly (wild cards) nella istruzione COPY.	Sono disponibili caratteri jolly che vi permettono di copiare TUTTI i file da un dischetto ad un altro.
Non è disponibile MEM.SAV.	È disponibile MEM.SAV, il che permette all'utente di avere più memoria. Questo rende il DOS più potente.
La gestione dei dischetti richiede 1 settore.	Il DOS II richiede 3 settori per poter gestire i drive in doppia densità. Come risultato i dischetti DOS I e DOS II non sono intercambiabili.
Non ha il file AUTORUN.SYS.	Ha il file AUTORUN.SYS che permette ai file di essere caricati ed eseguiti al caricamento.
File duplicati e copiati in un piccolo buffer.	Interi file duplicati o duplicati in un buffer che può essere grande come l'area di memoria riservata all'utente.
Non può unire 2 file.	Il SAVE BINARY FILE ha l'opzione "/A" che permette di unire 2 file.
Non ha il "carica e parti" (load-and-go).	Può creare file di tipo "carica e parti" (loand-and-go) che vi permettono di scegliere un file e farlo partire senza introdurre l'indirizzo RUN.
I settori danneggiati sul dischetto non vengono segnalati al formattamento.	Un dischetto con settori danneggiati non può essere formattato.
I margini non sono riposizionati automaticamente.	I margini sono rimessi alla posizione originale ogni volta che viene inserito DOS II.
Una sola versione di DOS I, sistema con unità a dischi in singola densità.	Due versioni di DOS II: sistema con singola e doppia densità o sistema solo con singola densità.
È necessario visualizzare il Menu prima di scegliere una nuova istruzione.	Possibile scelta fra visualizzare il Menu e inserire una nuova istruzione.
Può scrivere il file DOS solo nel Drive 1.	Può scrivere il file DOS su qualsiasi Drive.
Può avere solo tre file aperti simultaneamente.	Può avere fino a otto file aperti simultaneamente.
Non è disponibile il NOTE/POINT per l'accesso casuale.	È disponibile il NOTE/POINT per l'accesso casuale.

APPENDICE I

STRUTTURA DI UN FILE COMPOSTO

STRUTTURA DI UN FILE COMPOSTO USANDO IL C. COPY FILE

N° byte	N° dec.	N° esadec.	Descrizione
1	255	FF	Codice d'identificazione (Parte 1)
2	255	FF	
3	0	00	Indirizzo iniziale (Parte 1)
4	80	50	
5	31	1F	Indirizzo finale (Parte 1)
6	80	50	
.	.	.	DATI (Parte 1)
.	.	.	32 Byte
.	.	.	
38	255	FF	Codice d'identificazione (Parte 2)
39	255	FF	
40	32	20	Indirizzo iniziale (Parte 2)
41	80	50	
42	143	8F	Indirizzo finale (Parte 2)
43	80	50	
.	.	.	DATI (Parte 2)
.	.	.	112 Byte
.	.	.	

STRUTTURA DI UN FILE COMPOSTO USANDO K. BINARY SAVE

N° byte	N° dec.	N° esadec.	Descrizione
1	255	FF	Codice d'identificazione
2	255	FF	
3	0	00	Indirizzo iniziale (Parte 1)
4	80	50	
5	31	1F	Indirizzo finale (Parte 1)
6	80	50	
.	.	.	DATI (Parte 1)
.	.	.	32 Byte
.	.	.	
38	32	20	Indirizzo iniziale (Parte 2)
39	80	50	
40	143	8F	Indirizzo finale (Parte 2)
41	80	50	
.	.	.	DATI (Parte 2)
.	.	.	112 Byte
.	.	.	

APPENDICE J

GLOSSARIO DEI TERMINI

Accesso	Il metodo (o l'ordine) in cui le informazioni sono lette da, o scritte sul dischetto.
Alfanumerico (carattere)	Le lettere maiuscole da A a Z e i numeri da 0 a 9 e/o le combinazioni di lettere e numeri. Sono esclusi simboli grafici, punteggiatura e altri caratteri speciali.
Array	Un set unidimensionale di elementi cui è possibile far riferimento uno alla volta o come lista completa, usando il nome della variabile array e un indice. Così all'elemento 10 dell'array B si farà riferimento come B(10). Notate che non sono supportati array di stringhe dal BASIC, ma potete prelevare ciascun elemento di una stringa; per esempio A$(10). Tutti gli array devono essere dimensionati prima dell'uso. Una matrice è un array bidimensionale.
ATASCII	Il metodo di codifica utilizzato per memorizzare dati di testo. In ATASCII, che è una versione modificata dell'ASCII: l'American Standard Code for Information Interchange (Codice Standard Americano per l'Interscambio di Informazioni), ogni carattere e simbolo grafico, come la maggior parte dei tasti di controllo, ha un codice a un byte (0-255 decimale) che lo rappresenta. Vedere la tabella sul Manuale BASIC ATARI.
AUTORUN.SYS	Nome di un file riservato al Sistema Operativo su Disco (DOS).
Binary Load	Caricamento di un file oggetto in linguaggio macchina nella memoria del computer.
Binary Save	Salvataggio di file oggetto in linguaggio macchina in un'unità a dischi o un registratore di programmi.
Bit	Abbreviazione di "Binary Digit" (cifra binaria). La più piccola unità di informazione rappresentata dal valore 0 o 1.
Boot	È il programma di inizializzazione che "dispone" il computer quando viene acceso. Alla conclusione del Boot, il computer è in grado di caricare ed eseguire programmi a più alto livello.
Break	Per interrompere l'esecuzione di un programma. Premendo il tasto **BREAK** si causa un'interruzione nell'esecuzione.
Buffer	Un'area di memoria temporanea nella RAM usata per contenere dati per successive elaborazioni, operazioni di input/output e simili.
Byte	Otto bits. Un byte può rappresentare un carattere. Un byte varia da 0 a 255 (decimale).
Byte meno significativo	Il byte nella posizione più a destra in un numero o una parola.

Byte più significativo	Il byte più a sinistra in un numero o una parola.
CIO	Central Input/Output Subsystem (Sottosistema Centrale di Input/Output). La parte del sistema operativo che tratta l'input/output.
CLOSE	Per "chiudere" l'accesso a un file su disco. Prima di potervi nuovamente accedere, il file deve essere aperto ancora. Vedi OPEN.
Codice oggetto	Codice macchina derivato dal "codice sorgente"; tipicamente per il linguaggio Assembly.
Codice sorgente	Una serie di istruzioni, scritte in un linguaggio che non sia il linguaggio macchina, che quindi richiede una traduzione per essere eseguito.
Dati	Informazioni di qualsiasi tipo, generalmente un set di byte.
Debug	Isolare ed eliminare errori da un programma.
Decimale	Un sistema di numerazione che usa le cifre da 0 a 9. I numeri decimali sono memorizzati nel computer in formato decimale codificato in binario. Vedi Bit, Esadecimale e Ottale.
Default	(Valore sottinteso o valore mancante). Una condizione o valore definita dal computer in mancanza di indicazioni diverse da parte dell'utente. Per esempio, il computer si pone in GRAPHICS 0 finché non viene selezionato un altro modo grafico.
Delimitatore	Un carattere che segna l'inizio o la fine di un dato, ma non è parte dello stesso. Per esempio, le virgolette (") sono usate dalla maggior parte dei sistemi BASIC per delimitare stringhe.
Densità	La distribuzione di informazioni in una superficie di memoria, cioè il numero di byte per settore. In singola densità si registrano 128 byte per settore e in doppia densità 256 byte per settore.
Destinazione	La periferica o l'indirizzo che riceve il dato durante uno scambio di informazioni (specialmente uno scambio I/O). Vedere Sorgente.
Dischetto	Un piccolo disco, piatto e coperto da un involucro di protezione. Il tempo di accesso per un dischetto è molto minore che per un nastro.
Dischetto di sistema	Una coppia esatta del dischetto principale originale (Master Diskette). Usate sempre copie del dischetto principale invece dell'originale. Fate copie di tutti i dischetti di dati o programmi importanti.
DOS	Abbreviazione di Disk Operating System (Sistema Operativo su Disco): contiene il software e i programmi che facilitano l'uso del sistema di unità a dischi.
DOS.SYS	Nome di un file riservato al Sistema operativo su disco.
Esadecimale	Sistema di numerazione che usa 16 caratteri alfanumerici: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E ed F.
Estensione	Da 0 a 3 caratteri addizionali preceduti da un punto (necessario se si usa l'estensione) dopo il nome di un file (filename). Per esempio nel nome PHONLIST.BAS le lettere "BAS" rappresentano l'estensione.

File	Un insieme organizzato di dati correlati fra loro. Un file è il più grande insieme di informazioni a cui si può fare riferimento con un solo nome. Per esempio un programma BASIC viene memorizzato su un dischetto come file e vi si può accedere con le istruzioni SAVE o LOAD (fra le altre).
Fine file	Carattere convenzionale che informa il computer che è stata raggiunta la fine di un certo file su disco.
Formattare	Definire su un dischetto nuovo o cancellato le piste e i settori. Quando è formattato, ogni dischetto contiene 40 piste circolari con 18 settori per pista. Ogni settore può memorizzare fino a 128 byte di dati (densità singola) o 256 byte di dati (densità doppia).
Identificatore del Drive (Drivespec)	Parte dell'identificatore del file che dice al computer a quale drive accedere. Se è omesso, il computer assumerà che si tratti del Drive 1.
Identificatore del file (Filespec)	Una sequenza di caratteri che specifica un file su una particolare periferica. Non create due file con lo stesso identificatore; se lo fate e li memorizzate sullo stesso dischetto, non potrete accedere al secondo file. E, siccome essi hanno lo stesso identificatore, le funzioni DOS, DELETE FILE, RENAME FILE e COPY FILE agiranno su entrambi.
Indirizzamento casuale	Il metodo di accesso ai dati su un dischetto direttamente dal byte e settore dove sono memorizzati, senza leggere l'intero file sequenzialmente.
Indirizzamento indicizzato	Vedi accesso casuale.
Indirizzamento sequenziale	Il metodo di accesso ad ogni byte sul dischetto in ordine, partendo dal primo byte del settore.
Indirizzo	Una locazione di memoria, specificata generalmente con un numero a due byte in formato decimale o esadecimale. (Varia da 0 ad un massimo di FFFF esadecimale).
Input	Il trasferimento di dati dall'esterno del computer (ad esempio un file su dischetto) alla RAM. L'output è l'inverso, e le due parole sono spesso usate insieme per descrivere le operazioni di trasferimento dati: Input/Output o semplicemente I/O. Notate che il punto di riferimento è sempre il computer, così Input significa: nel computer.
INPUT	Istruzione BASIC usata per richiedere sia dati numerici che stringhe da una specifica periferica.
iocb	Una espressione aritmetica il cui risultato è un numero da 1 a 7. Tale numero è usato per fare riferimento ad un file o ad una periferica.
IOCB	Input/Output Control Block (Blocco di Controllo Input/Output). Una sezione della RAM riservata alle operazioni di I/O con una periferica.
Kilobite o K.	1024 byte. Quindi una memoria di 16K RAM ci dà effettivamente 16×1024 cioè 16.384 byte.
Linguaggio Macchina	Il set d'istruzioni del microprocessore usato, nel caso dell'ATARI il 6502.

Lista dei file (Directory)	Un elenco dei file contenuti su un dischetto contenente i nomi e le dimensioni.
Nome del file (Filename)	I caratteri alfanumerici usati per identificare un file. Possono essere usati un totale di otto numeri e/o lettere, la prima deve essere una lettera.
Numero del Drive	Un numero da 1 a 8 che specifica quale drive deve essere usato.
OPEN	Prepara un file per potervi accedere, specificando se sarà eseguita una operazione di input o di output e dichiarando l'identificatore del file.
Ottale	Sistema di numerazione che usa le cifre da 0 a 7. Indirizzi e valori di byte sono a volte dati in forma ottale.
Parametro	Variabile in una istruzione o funzione.
Periferica	Un dispositivo I/O.
Pista	Un cerchio su un dischetto usato per la memorizzazione magnetica di dati. Ogni pista ha 18 settori, ognuno dei quali ha una capacità di memorizzazione di 128 byte. Ci sono un totale di 40 piste su ogni dischetto.
POKEY	Un chip I/O che gestisce le comunicazioni sui bus seriali.
Protezione (Write protect)	Un metodo per impedire all'unità a dischi di scrivere su un dischetto. I dischetti ATARI vengono protetti coprendo con una piccola etichetta una tacca sul dischetto.
Puntatore del file	Un puntatore ad una locazione del file. Ogni file ha il suo puntatore.
Punto d'ingresso	L'indirizzo dal quale una routine o un programma in linguaggio macchina devono iniziare (indirizzo di trasferimento).
Record	Un blocco di dati delimitati dal carattere 9B esadecimale (fine linea).
Settore	Un settore è il più piccolo blocco di dati che può essere scritto o letto su un file. Ogni settore a densità singola può memorizzare 128 byte di dati e ogni settore a doppia densità ne può memorizzare 256.
Sorgente	La periferica o indirizzo che contiene i dati da inviare ad una destinazione. Vedi destinazione.
Stringa	Una sequenza di lettere o caratteri memorizzati in una variabile stringa. Il nome della variabile stringa deve terminare con $.
Stringa Nulla	Una stringa che non contiene caratteri. Per esempio, A$="" memorizza la stringa nulla in A$.
Tokenizzare	Il processo d'interpretazione del testo sorgente BASIC e di conversione di esso nel formato interno usato dall'interprete BASIC.
Variabile	Una variabile deve essere pensata come una scatola nella quale può essere memorizzato un valore. Tali valori sono tipicamente numeri e stringhe.
Velocità (Band rate)	Velocità di trasmissione segnali in bit al secondo.

INDICE ANALITICO

K

L

M

N

O

P

R

S

T

U

V

W

N° ERRORE	MESSAGGIO DEL CODICE D'ERRORE
2	Memoria insufficiente
3	Errore nel valore
4	Troppe variabili
5	Errore nella lunghezza di una stringa
6	Fuori dai DATA
7	Numero maggiore di 32767
8	Errore nell'istruzione Input
9	Errore nel DIM
11	Overflow/Underflow in virgola mobile
12	Linea non trovata
13	NEXT senza il FOR
15	GOSUB o FOR cancellati
16	Errore sul RETURN
17	Errore di sintassi
18	Errore nella funzione VAL
19	LOAD di un programma troppo lungo
20	Numero di periferica > 7 o = 0
21	Errore nel LOAD di un file
128	Interruzione causata da BREAK
129	IOCB* già aperto
130	Periferica specificata non esistente
131	IOCB a sola scrittura
132	Istruzione scorretta
133	Periferica o file non aperto
134	Numero IOCB scorretto
135	IOCB a sola lettura
136	Fine del file
137	Record troncato
138	Periferica fuori tempo
139	Periferica non riconosciuta
141	Cursore fuori dai limiti per quel particolare modo grafico
142	Dati seriali in eccesso
143	Errore del cheksum nei dati seriali
144	La periferica non esegue
145	Modo grafico scorretto
146	Funzione non implementata
147	RAM insufficiente per il modo grafico selezionato
160	Errore nel numero del drive
161	Troppi file aperti (non è disponibile il buffer di settore)
162	Disco pieno (non ci sono settori disponibili)
163	Errore di sistema irrecuperabile nei dati di I/O
164	Numero di file errato
165	Errore nel nome del file
166	Errore nella lunghezza del dato POINT
167	File bloccato (Locked)
168	Istruzione scorretta
169	Lista dei file (Directory) piena (64 file)
170	File non trovato
171	POINT non valido
172	Tentativo di unire un file DOS I usando DOS II
173	Settori danneggiati al formattamento

* IOCB si riferisce ai Blocchi di controllo INPUT/OUTPUT.

IMPORTANTE

Come tutte le apparecchiature elettriche, l'Home Computer ATARI e le apparecchiature ad esso collegate, utilizzano energia nella gamma delle frequenze radio.
Le apparecchiature ATARI sono state costruite in modo da non costituire una sorgente di interferenza fastidiosa durante il loro utilizzo in un ambiente domestico.
Per minimizzare o prevenire eventuali disturbi alla ricezione radio-televisiva, vengono proposti dei semplici suggerimenti.
Spegnere e riaccendere le apparecchiature controllando se, con le apparecchiature spente, i disturbi spariscono. Se i disturbi fossero presenti solo durante il funzionamento delle apparecchiature, eseguire una o più delle seguenti operazioni:

- orientare l'antenna della radio e della televisione in un modo diverso
- cercare una sistemazione alternativa alle apparecchiature rispetto alla radio o alla televisione
- distanziare maggiormente le apparecchiature dalla radio o dalla televisione
- collegare le apparecchiature ad una presa che non sia su di un ramo comune alla radio e alla televisione

Se necessario richiedere assistenza al rappresentante ATARI più vicino oppure ad un tecnico radio-televisione qualificato.

A Warner Communications Company